



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

MODELAMIENTO Y SIMULACION DE UN ALGORITMO PARA EL CONTROL DEL BRAZO ROBÓTICO.

IRMA CATALINA VILLA ESCUDERO

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la **ESPOCH**, como requisito parcial para la obtención del grado de:

**MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN
INDUSTRIAL**

Riobamba – Ecuador

Julio 2017



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El **Trabajo de Titulación** modalidad **Proyectos de Investigación y Desarrollo**, titulado "**MODELAMIENTO Y SIMULACIÓN DE UN ALGORITMO PARA EL CONTROL DEL BRAZO ROBÓTICO**", de responsabilidad de la Ing. Irma Catalina Villa Escudero, ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

Ing. Oswaldo Martínez Guashima; M.Sc
PRESIDENTE

FIRMA

Ing. Douglas Plaza Guingla, Ph.D
DIRECTOR

FIRMA

Ing. Miguel Pérez Bayas M.Sc.
MIEMBRO

FIRMA

Ing. Javier Gavilanes Carrión M.Sc.
MIEMBRO

FIRMA

Riobamba, Julio 2017

DERECHOS INTELECTUALES

Yo, Irma Catalina Villa Escudero, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, y el patrimonio intelectual generado por la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

IRMA CATALINA VILLA ESCUDERO

N° Cédula: 0603607359

DEDICATORIA

Dedico este proyecto a Dios por haberme dado la vida y permitirme el haber llegado hasta este momento significativo en mi vida profesional. A mis padres y hermanos quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, con sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y *capacidad*. Es por ello que soy lo que soy ahora. Los amo con mi vida.

Katty

AGRADECIMIENTO

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida de llena de aprendizajes, experiencias y sobre todo felicidad.

A mis padres y hermanos que han dado todo el esfuerzo para que ahora este culminando esta etapa de mi vida y darles las gracias por apoyarme en todos los momentos difíciles de mi vida tales como la felicidad la tristeza pero ellos siempre han estado junto a mí.

Son muchas las personas que han formado parte de mi vida profesional a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.

Katty

ÍNDICE DE CONTENIDOS

RESUMEN.....	xii
SUMMARY	xiii
CAPÍTULO I	
PROBLEMA DE INVESTIGACIÓN.....	1
1.1. Planteamiento del problema.....	1
1.2. Formulación del problema	1
1.3. Sistematización del problema	1
1.4. Justificación de la investigación.....	2
1.5. Objetivos de la investigación	3
1.5.1. <i>Objetivo general</i>	3
1.5.2. <i>Objetivos específicos</i>	3
1.6. Hipótesis.....	3
CAPÍTULO II	
2. ESTADO DEL ARTE.....	4
2.1. Antecedentes de investigación	4
2.2. Definiciones	7
2.2.1. <i>Robot</i>	7
2.2.2. <i>Robot manipulador</i>	7
2.2.3. <i>Tipos de Robots</i>	7
2.2.4. <i>Partes de un robot</i>	8
2.2.5. <i>Eslabón</i>	8
2.2.6. <i>Articulación</i>	9
2.2.7. <i>Variable articular</i>	9
2.2.8. <i>Grados de libertad</i>	9
2.2.9. <i>Configuraciones morfológicas de los robots</i>	9
2.2.10. <i>Cinemática</i>	10
2.2.11. <i>Algoritmos y movimiento</i>	11
2.2.12. <i>Denavit – Hartenberg (D-H)</i>	12
2.2.13. <i>Simulación</i>	13
2.3. El sistema robótico kawasaki RS 003N	15
2.3.1. <i>Características del Robot Kawasaki RS003N</i>	15
2.3.2. <i>Especificaciones técnicas del Robot Kawasaki RS003N</i>	16
2.3.3. <i>Generalidades del sistema</i>	17

2.3.4.	<i>Módulo de Control</i>	17
2.3.5.	<i>Actuadores</i>	18
2.4.	Software empleado para la simulación y control y de un robot manipulador	18
2.4.1.	<i>SolidWorks</i>	18
2.4.2.	<i>Matlab</i>	19
2.4.3.	<i>Componentes de Matlab</i>	20
2.4.4.	<i>Simulink</i>	22
2.4.5.	<i>Bibliotecas fundamentales de SimMechanics</i>	22
CAPÍTULO III		
3.	DISEÑO DE INVESTIGACIÓN.....	26
3.1.	Modalidad de la Investigación	26
3.2.	Tipos de Investigación	26
3.3.	Métodos de investigación.....	26
3.4.	Técnicas e Instrumentos de recolección de datos.....	26
3.5.	Diseño del proyecto.....	27
3.6.	Presupuesto	27
CAPÍTULO IV		
4.	SIMULACIÓN Y CONTROL DEL BRAZO ROBÓTICO KAWASAKI RS 003N	28
4.1.	Análisis del Robot Kawasaki RS003N	28
4.1.1.	<i>Mecanismos del robot</i>	28
4.2.	<i>Diagrama del proyecto</i>	30
4.3.	Requerimientos del Sistema.....	30
4.4.	<i>Diagrama de Conexión del Modelo</i>	31
4.5.	Modelado del Brazo Robótico utilizando Solidworks de CAD.	32
4.6.	Diseño de la Estructura del brazo robótico en Simulink por medio de SimMechanics.	38
4.6.1.	<i>Bloques Usados de SimMechanics</i>	39
4.7.	Exportación de SolidWorks a Simulink	41
4.8.	Modelado del Sistema de Control	45
4.8.1.	<i>Control del movimiento del robot sin el uso de algoritmo</i>	45
4.8.2.	<i>Control del movimiento del robot mediante el algoritmo Denavit - Hartenberg</i>	48
4.8.3.	<i>Parametros de Denavit - Hartenberg</i>	48
4.8.3.1.	<i>Ubicación de la variables articulares</i>	48
4.8.3.2.	<i>Modelo cinemático directo usando la librería de Peter Corke</i>	49
4.8.3.3.	<i>Control del movimiento del robot a través de cinemática directa</i>	50

4.8.3.4.	<i>Control del movimiento del robot a través de cinemática inversa.....</i>	52
4.9.	PRUEBAS DE POSICIONAMIENTO.....	57
4.10.	RESULTADOS Y DISCUSIÓN.....	61
4.1.	Comprobación de Hipótesis	61
	CONCLUSIONES	64
	RECOMENDACIONES	65
	BIBLIOGRAFIA	
	ANEXOS	

LISTA DE TABLAS

Tabla 1-2: Tipos de Robots.....	8
Tabla 2-2: Especificaciones técnicas del Robot Kawasaki RS003N	16
Tabla 3-2: Requerimientos para el funcionamiento de Solidworks	18
Tabla 4-2: Requisitos adicionales de Matlab	20
Tabla 1-3: Presupuesto.....	27
Tabla 1-4: Clasificación de los robots según la AFRI	29
Tabla 2-4: Ángulos máximos y mínimos establecidos para las juntas	45
Tabla 3-4: Cuadro de posiciones ejecutadas con la planta física y la simulación.....	57

INDICE DE FIGURAS

Figura 1-2: Tipos de Robots manipuladores	10
Figura 2-2: Diagra de relación entre cinemática directa e inversa	11
Figura 3-2: Robot Kawasaki RS003N.....	15
Figura 4-2: Bibliotecas de SimMechanics	23
Figura 5-2:Biblioteca Bodies	23
Figura 6-2: Biblioteca Joints	24
Figura 7-2: Biblioteca Sensors & Actuators	25
Figura 1-4: Esquema con la ubicación de los ejes del Robot Kawasaki RS003N	28
Figura 2-4: Rango y dimensiones del Robot Kawasaki RS 003N	29
Figura 3-4: Diagrama del Proyecto	30
Figura 4-4: Diagrama de Conexión del Modelo.....	31
Figura 5-4: Diseño de la base del brazo robótico.....	32
Figura 6-4: Diseño del cuerpo 1 del brazo robótico.....	33
Figura 7-4: Diseño del cuerpo 2 del brazo robótico.....	33
Figura 8-4: Diseño del cuerpo 3 del brazo robótico.....	34
Figura 9-4: Diseño del cuerpo 4 del brazo robótico.....	34
Figura 10-4. Diseño del cuerpo 5 del brazo robótico.....	35
Figura 11-4: Diseño de la mordaza del brazo robótico	35
Figura 12-4: Diseño de la pinza del brazo robótico	36
Figura 13-4: Ensamblaje de la pinza en el brazo robótico	36
Figura 14-4: Vista en 3D de la pinza ensamblada completa	37
Figura 15-4: Vista del brazo robótico completo.....	37
Figura 16-4: Vista en 3D de los diferentes ángulos del brazo robótico	38
Figura 17-4: Bloques Solid de las piezas del Kawasaki RS 003N.....	40
Figura 18-4: Bloques de la Librería SimMechanics.....	40
Figura 19-4: Exportación de Solidworks a Simulink.....	41
Figura 20-4: Importación a Matlab	42
Figura 21-4: Generación del modelado en Simulink.....	43
Figura 22-4: Modelo del control de las Revolute y Prismatic con Join Actuator	43
Figura 23-4: Parámetros de los bodys	44
Figura 24-4: Aspecto de la animación 3D del simulador del robot Kawasaki	44

Figura 25-4: Modelado del sistema de control de los ángulos del robot.....	46
Figura 26-4: Despliegue del movimiento de las juntas 3 y 5.....	47
Figura 27-4: Despliegue del movimiento de pinza (juntura 7)	47
Figura 28-4: Variables articulares del robot Kawasaki RS 003N	49
Figura 29-4: Modelo cinemático directo usando la librería de Peter Corke.	49
Figura 30-4: Visualización del robot en diagrama de bloques.....	53
Figura 31-4: Modelo cinemático directo del movimiento del robot	54
Figura 32-4: Control cinemático directo usando la librería de Peter Corke.....	55
Figura 33-4: Modelo cinemático inverso del movimiento del robot.....	56
Figura 34-4: Control cinemático inverso usando la librería de Peter Corke	56
Figura 35-4: Posicionamiento del robot físico y en 3D (posición 0)	57
Figura 36-4: Posicionamiento del robot físico y en 3D (posición 1)	58
Figura 37-4: Posicionamiento del robot físico y en 3D (posición 2)	58
Figura 38-4: Posicionamiento del robot físico y en 3D (posición 3)	59
Figura 39-4: Posicionamiento del robot físico y en 3D (posición 4)	59
Figura 40-4: Posicionamiento del robot físico y en 3D (posición 5)	60
Figura 41-4: Posicionamiento del robot físico y en 3D (posición 6)	60
Figura 42-4: Modelado de piezas en Solid Works	62
Figura 43-4: Robot Kawasaki RS 003N en 3D.....	62

ÍNDICE DE ANEXOS

ANEXO A: MODELADO CINEMÁTICO DE ROBOT KAWASAKI RS 03N.....	65
ANEXO B: ANEXO B: DIAGRAMA DE FLUJO DE LA SIMULACIÓN.....	71

RESUMEN

Se desarrollo el modelo cinemático, para el control y la simulación, mediante el ordenador al brazo robótico KAWASAKI RS 003N, que se localiza en la Facultad de Mecánica de la ESPOCH. Se realizó el diseño y modelamiento del brazo robótico utilizando Solidworks y CAD, posteriormente se ha efectuado la importación del Modelo CAD a Simulink, a la par se ha definido la estructura mecánica del brazo robótico en Simulink por medio de la herramienta SimMechanics, posterior se desarrollo el algoritmo de control para insertar las trayectorias al manipulador y finalmente se realizaron pruebas de simulación así como el control de los movimientos del brazo robótico utilizando los software Matlab y Simulink. Dentro de los resultados obtenidos el algoritmo Denavith Hartenberg, permite el adecuado control de los movimientos del robot. La interfaz gráfica tanto del diagrama de líneas como del robot en 3D de Simulink muestra adecuadamente el posicionamiento y la orientación del robot de acuerdo con las órdenes emitidas por el usuario en los cuadros de mando, logrando visualizar en pantalla las posiciones de cada uno de los eslabones del robot si como la apertura y cierre de la pinza. En cuanto al proceso de exportación es conveniente la instalación previa de la herramienta SimMechanics Link, para limitar la presencia de fallos en el sistema.

Palabras clave:

<TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <CONTROL AUTOMÁTICO>,
<ROBÓTICA>, <SIMMECHANICS (SOFTWARE)>, <SOLIDWORKS (SOFTWARE)>,
<MATLAB (SOFTWARE)>, <DENAITH HARTENBERG (ALGORITMO)>.

SUMMARY

The kinematic model was developed for the monitoring and simulation by the computer to the KAWASAKI RS003N robotic arm, which is located at ESPOCH in the Faculty of Mechanics. The design and modelling of the robotic arm was developed by using SolidWorks and CAD (Computer-aided Design), afterwards, the CAD model was imported to Simulink, while the mechanical structure of the robotic arm in Simulink was defined by means of the SimMechanics tool. After this process, the control algorithm was developed in order to insert the trajectories to the manipulator, and finally, simulation tests were conducted as well as the control of the movements of the robotic arm by using Matlab and Simulink software. The Denavit-Hartenberg in the findings reported, makes possible the appropriate control of the movements of the robot. The graphical interface of both, the line diagram and Simulink 3D robot, shows appropriately the positioning and orientation of the according to the commands sent by the user in the control panels, being able to display on the screen the positions of the each of the Robot links as the opening and closing of the gripper. Regarding to the exportation process it is advisable to install the SimMechanics Link previously in order to reduce failures in the system.

Keywords:

TECHNOLOGY AND ENGINEERING SCIENCES, AUTOMATIC CONTROL, ROBOTICS, SIMMECHANICS (SOFTWARE), SOLIDWORKS (SOFTWARE), MATLAB (SOFTWARE), DENAVIT-HARTENBERG (ALGORITHM)

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

La creciente aparición de robots; mecanismos controlados por elementos eléctricos y electrónicos, capaces de desarrollar tareas industriales previamente dispuestas por el ser humano mediante un lenguaje de programación; requiere la implementación de sistemas de control que accionen y controlen la ejecución de dichas tareas.

La Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo, registra dentro de sus equipos a KAWASAKI RS 003N, un brazo robótico de la industria japonesa Kawasaki Robotics, el cual aún no ha sido programado, por ello se pretende buscar una alternativa para su manipulación y control.

La implementación de un sistema de control para un robot resulta ser un problema complejo, debido a que sus propiedades cinemáticas y dinámicas varían de acuerdo con la posición. Actualmente, gracias al avance tecnológico existe una serie de herramientas de hardware y software que permiten el modelamiento y control automático de sistemas en tiempo real.

Se requiere entonces la realización del modelamiento y simulación de un algoritmo de control para el robot KAWASAKI RS 003N, para cuyo propósito se pretende la utilización de Solid Works, Matlab y Simulink.

1.2. Formulación del problema

¿De qué manera debe realizarse la manipulación y control del Brazo Robótico KAWASAKI RS 003N?

1.3. Sistematización del problema

❖ *¿Cuál es el funcionamiento integral del Brazo Robótico KAWASAKI RS 003N?*

- ❖ *¿Cuáles son las características de SolidWorks, Matlab y Simulink y otros elementos necesarios para el control de los movimientos del brazo robótico?*
- ❖ *¿Cómo debe efectuarse la importación del modelo matemático al simulador?*
- ❖ *¿De qué manera debe establecerse la estructura mecánica del brazo robótico para probar el sistema de control?*
- ❖ *¿Cómo debe desarrollarse el algoritmo de control para insertar las trayectorias al manipulador?*
- ❖ *¿De qué forma debe realizarse el Control y Simulación del movimiento del brazo robótico?*

1.4. Justificación de la investigación

El Octavo Objetivo del Milenio se orienta a fomentar asociaciones para el desarrollo, a demás pretende la colaboración entre los sectores público, privado y la universidad a fin de aprovechar los beneficios de la tecnología. Por otro lado, de acuerdo con el Plan Nacional del Buen Vivir, en el marco de la estrategia de acumulación, distribución y redistribución, también se precisa del desarrollo de las fuerzas productivas que conlleven al fomento de los sectores priorizados para la transformación de la matriz productiva a mediano y largo plazo.

La articulación colaborativa entre los sectores público, privado y la universidad, permite que el sector de conocimiento y educación proponga mecanismos eficientes de traducción del conocimiento en la generación de nuevos productos y medios de producción que satisfagan las necesidades y que fomenten las capacidades de la población ecuatoriana. (Senplades, 2013)

Desde esta perspectiva, el propósito de esta investigación es realizar el Modelamiento y Simulación de un Algoritmo para el Control del Brazo Robótico, utilizando la funcionalidad de Solid Works, Matlab y Simulink para el control del movimiento. El complemento SimMechanics de Simulink en Matlab nos permite la importación de un sistema mecánico ya modelado en un software CAD 3D, proporcionándonos el sistema ya representado en bloques en Simulink para la manipulación y control del brazo robótico en una forma sencilla y versátil.

Por lo expuesto, este trabajo investigativo se justifica teóricamente en cuanto se emplean las diversas teorías conocidas durante los años de estudio y las revisadas durante la investigación, mismas que sustentan todo el proceso investigativo (Infante & Ortíz, 2004). También se justifica metodológicamente, debido a que todo el proceso de investigación se apoya en el método científico.

Finalmente, nuestra investigación se justifica empíricamente porque en ella se ponen en práctica todas las teorías y conocimientos científicos, a fin de proponer alternativas viables de solución a la problemática identificada.

1.5. Objetivos de la investigación

1.5.1. Objetivo general

Realizar el Modelamiento y Simulación de un Algoritmo para el Control del Brazo Robótico.

1.5.2. Objetivos específicos

- ❖ Realizar el Modelamiento CAD del Brazo Robótico utilizando Solidworks.
- ❖ Efectuar la importación del Modelo CAD a Simulink.
- ❖ Definir la estructura mecánica del brazo robótico en Simulink por medio de la herramienta SimMechanics para probar el sistema de control.
- ❖ Desarrollar el algoritmo de control para insertar las trayectorias al manipulador.
- ❖ Realizar el Control y Simulación del movimiento del brazo robótico utilizando Matlab y Simulink.

1.6. Hipótesis

Es posible realizar el modelamiento, simulación y control de los movimientos del Brazo Robótico KAWASAKI RS 003N, utilizando Solid Works, Simulink de Matlab y el Algoritmo Denavit – Hartenberg.

CAPÍTULO II

2. ESTADO DEL ARTE

2.1. Antecedentes de investigación

La tesis denominada “MODELAMIENTO Y SIMULACIÓN DE UN ALGORITMO DE CONTROL PARA EL BRAZO ROBÓTICO”, realizada por el Egresado de Escuela Politécnica Nacional de Ecuador, David Alejandro Jaramillo Rojas; concluye que es comprobada la practicidad del uso de Simulink – SimMechanics para el estudio de la cinemática y dinámica de mecanismos; y, que las rutinas creadas para la simulación pueden ser aplicadas para la construcción de un prototipo controlado con la plataforma Matlab en tiempo real.

Por otro lado el trabajo de investigación: “DISEÑO Y SIMULACIÓN DE UN ALGORITMO PARA EL CONTROL DE UN ROBOT MODULAR TIPO CADENA”, elaborado por Rodrigo Moreno García, Maestrante de la Universidad Nacional de Colombia; presentó una estrategia híbrida de control para un robot modular tipo cadena. El modelo se basa en la combinación de dos técnicas bioinspiradas como son los CPG y la comunicación por medio de mensajes inspirados en hormonas. La estrategia de control se guía bajo el paradigma de formar comportamientos globales complejos a partir de interacciones locales simples, es así como se utiliza un modelo de osciladores de fase con amplitud controlada para generar movimientos coordinados en una estructura. Se recalca la utilidad de la estrategia de control empleada.

De la tesis titulada: “MODELADO, IDENTIFICACIÓN Y CONTROL DE UN ROBOT MANIPULADOR DE DOS GRADOS DE LIBERTAD”, realizada por Ernesto Javier Moreno Valenzuela, Egresado del Instituto Politécnico Nacional de México; en la que se trata el problema de modelado, identificación y control de un robot manipulador experimental de dos grados de libertad; se observa que la identificación de los parámetros del robot es llevada a cabo usando el modelo dinámico de la regresión filtrado y los métodos de mínimos cuadrados ponderado y mínimos cuadrados recursivo. Se presentan resultados experimentales del control de seguimiento de trayectorias con el robot manipulador, utilizando diferentes esquemas de control basados en el modelo, validando de esta manera la técnica aplicada.

En cuanto al trabajo de investigación: “CONTROL AUTOMÁTICO DE UN BRAZO ROBOT DE 5 GRADOS DE LIBERTAD CON ARDUINO”, ejecutado por el Egresado de la Universidad de Valladolid, Juan José Yagüe Niño. Cuyo propósito es desarrollar un sistema completo de automatizado extrapolable a cualquier otro robot; dotando al mismo de un sistema para leer la posición de los motores y creando un programa de control y generación de trayectorias. El interfaz de control se diseña a partir de un controlador Arduino Mega 2560, el software de control y comunicación se desarrolla a partir de la plataforma .NET de Microsoft. Arduino controla el movimiento automatizado y los límites de movimiento de cada eje y almacena la información de programación de movimientos. El control automático; la programación de trayectorias; los controles de parada automática; y, el resto de funcionalidades típicas de control se realizan a través del software .NET. Se concluye que una de las “limitaciones” de Arduino en relación a otros lenguajes de programación, es la inexistencia del tipo de datos “String”, lo cual supone un inconveniente a la hora de recibir los parámetros de configuración de trayectorias.

El estudio denominado: “MODELO CINEMÁTICO Y CONTROL DE UN BRAZO ROBÓTICO IMPRIMIBLE”, realizado por Juan Carlos Rodríguez Zambrana, Egresado de la Universidad Carlos III de Madrid; cuyo objetivo es el desarrollo del modelo cinemático, el control y la simulación de un brazo robótico imprimible, mediante un ordenador; para lo cual el problema de la cinemática directa ha sido solucionado a través del algoritmo Denavit – Hartenberg; por otro lado se ha implementado un programa en C++ y OpenGL, el cual incorpora una interfaz gráfica que permite la representación en tres dimensiones de la posición y orientación del brazo robótico. En las pruebas de funcionamiento se muestra que el usuario envía una orden a través del gamepad, se simulan los movimientos en la pantalla y, finalmente, se ejecuta la orden en el robot real, validando de esta manera la practicidad del uso del algoritmo Denavit – Hartenberg.

La investigación titulada: “MODELADO DINÁMICO Y SIMULACIÓN DEL ROBOT INDUSTRIAL STÄUBLI TX90”, desarrollada por Fernando Mazo Espiau, Egresado de Escuela Técnica Superior de Ingeniería Industrial de Barcelona; parte de una aproximación mediante técnicas CAD con el programa Solid Works y luego se aplican dos técnicas de identificación directa, evaluando las ecuaciones del movimiento con datos obtenidos directamente del robot. Se ha implementado un programa en C que permite generar trayectorias parametrizadas como series de Fourier finitas. Para realizar el modelo dinámico se probaron los métodos Lagrange y Newton-Euler, observando que los dos llegan al mismo resultado, sin embargo, las ecuaciones de Newton-

Euler comprenden menos términos y por lo tanto con su aplicación se minimiza el tiempo de cómputo con relación a Lagrange. Se recomienda el uso de Solid Works para el modelado CAD.

El trabajo de investigación: “DISEÑO DE UN SIMULADOR 3D PARA ROBOTS MANIPULADORES DE 3 GRADOS DE LIBERTAD”, realizado por Bravo Hernández Paola, Reyes Cortes Fernando, Vergara Limón Sergio y Ríos Silva Eduardo; de la Universidad Autónoma de Puebla; muestra el desarrollo del software SimRoMa (Simulador de Robots Manipuladores), bajo el entorno de Windows, realizando el programa con el compilador de Visual C++. Dentro de las aportaciones de la investigación se proporciona el desarrollo del software para la creación del simulador gráfico en 3D, destacando que no solo es una animación sino la simulación de los fenómenos físicos. Se concluye que con la simulación del comportamiento del robot, se satisface la necesidad de controlar las variables y analizar diversos controladores para llegar a la posición deseada del robot.

Finalmente, la tesis denominada: “DESARROLLO DE ESTRATEGIA DE CONTROL PARA UN BRAZO MECÁNICO REPARADOR DE TUBERÍAS HIDROSANITARIAS, efectuada por Julián David Pareja Garzón, de la Pontificia Universidad Javeriana de Colombia; evidencia que se puede hacer un control mediante Gain Scheduling, teniendo especial cuidado con el método que se utiliza para hacer el control en cada punto de trayectoria, pues si el método usado para calcular el control no genera valores de K uniformes, se tienen problemas de interpolación causando que el control LPV sea inestable. Además recomienda primero modelar el robot en el programa Solid Works y luego mediante la herramienta Sim Mechanics pasarlo al entorno de MATLAB.

De los antecedentes de investigación se puede concluir que es conveniente y válido:

- ❖ Solid Works para el desarrollo de piezas, ensambles y planos de gran variedad de sistemas y mecanismos.
- ❖ Matlab para realizar cálculos técnicos científicos, pues integra operaciones de cálculo, visualización, programación y permite la interacción con el usuario.
- ❖ Simulink para construir y simular modelos de sistemas físicos y sistemas de control.
- ❖ SimMechanics para representar las piezas del sistema, articulaciones y elementos actuantes, también permite la importación de modelos CAD 3D ya desarrollados.
- ❖ El algoritmo Denavit - Hartenberg para establecer de manera sistemática un sistema de coordenadas.

Por lo expresado, en la presente investigación se considera el uso de Solid Works para el modelado CAD del robot; SimMechanics de Simuink para la simulación de los movimientos y el Algoritmo Denavit - Hartenberg para el control automático de los mismos.

2.2. Definiciones

2.2.1. Robot

La British Robot Association manifiesta que robot industrial es: “un dispositivo reprogramable destinado tanto a manipular piezas, herramientas o útiles específicos de fabricación mediante movimientos variables susceptibles de ser programados, para realizar operaciones de fabricación determinadas”; (González, Pineda, & Soberanes, 2007)

“El robot industrial es un “manipulador multifuncional reprogramable, capaz de manipular materias, piezas, herramientas o dispositivos especiales, según trayectorias variables programadas para realizar tareas diversas”. (González & Jimenez, 2008).

Cabe recalcar que el término robot puede referirse tanto a mecanismos físicos como a sistemas virtuales de software.

2.2.2. Robot manipulador

Se puede describir un robot manipulador como un conjunto de eslabones articulados que le permiten adoptar un número de posiciones diferentes para alcanzar ciertos objetivos en el espacio tridimensional. (Gómez, 2005)

2.2.3. Tipos de Robots

Existen tres grandes grupos de robots: móviles, industriales y humanoides.

Tabla 1-2: Tipos de Robots

Robots móviles	Robots Industriales	Robots Humanoides
Son cadenas cinemáticas abiertas y alternativamente abiertas y cerradas en el caso de robots con patas.	Son generalmente cadenas cinemáticas abiertas provistas de una base y un efector final.	Son cadenas cinemáticas que combinan las características de robots móviles con patas y manipuladores.
Empleados mayormente en ambientes académicos y de investigación.	Empleados en ambientes industriales y de producción.	Empleados en ambientes sociales como guías turísticos, asistente de niños y ancianos etc.
Disponen de sistemas sensoriales mas sofisticados.	Su sistema sensorial se limita al control de precisión del movimiento de sus articulaciones.	El sistema sensorial utiliza sistemas avanzados de Inteligencia Artificial, como visión y reconocimiento de voz.
Su cinemática y dinámica suele ser simple y limitada a cuatro tipos de configuración.	Su cinemática y dinámica puede ser compleja dependiendo de sus articulaciones.	La cinemática y dinámica de estos robots se concentra en la estabilidad y equilibrio al caminar, correr o saltar.
Emplean sistemas electrónicos en su sistema de actuadores.	Pueden emplear electrónica, neumática, hidráulica en su sistema de actuadores.	Emplean una combinación de actuadores electrónicos neumáticos e hidráulicos en su sistema de actuación.

Fuente: (Almeida, 2009)

2.2.4. Partes de un robot

Consta de elementos mecánicos, componentes eléctricos, electrónicos, neumáticos e hidráulicos, mismos que le permiten realizar diversas tareas, de acuerdo con las trayectorias variables programadas o con los programas preestablecidos.

2.2.5. Eslabón

Se refiere a cada uno de los segmentos rígidos que componen al robot.

2.2.6. *Articulación*

Representa la unión entre dos eslabones, cuya particularidad es permitir cierto movimiento relativo entre ellos. La robótica emplea primordialmente dos tipos de articulaciones rotacional y prismática.

2.2.7. *Variable articular*

Se refiere a la peculiaridad de la articulación de tomar cualquier valor sea en grados o en centímetros, valor que determina la posición específica de un eslabón con respecto al otro.

2.2.8. *Grados de libertad*

El eslabón y articulación en su conjunto le otorgan al robot un grado de libertad, lo que hace que los grados de libertad coincidan con el número de articulaciones o juntas.

2.2.9. *Configuraciones morfológicas de los robots*

De acuerdo con la geometría de su estructura mecánica, los robots manipuladores se clasifican:

- ❖ **Cartesiano.** Es un robot de alta precisión y velocidad, posee tres ejes de movimiento lineal perpendiculares entre sí, amplia zona de trabajo planar y un sistema de control sencillo.
- ❖ **Cilíndrico.** Presenta una articulación rotacional sobre la base y dos articulaciones lineales perpendiculares, lo cual le permite realizar movimientos en altura y en radio, presenta una mejor maniobrabilidad y velocidad que el robot cartesiano.
- ❖ **Polar.** Este manipulador posee dos articulaciones rotacionales y una lineal. Su accesibilidad es mejor que la del robot cartesiano y cilíndrico.
- ❖ **Esférico o articulado.** Posee tres articulaciones rotacionales. Han sido sustituidos por los robots de configuración angular.

- ❖ **SCARA** (*Selective Compliance Assembly Robot Arm*). Cuenta con varios tipos de articulaciones, son robots muy rápidos y de muy alta precisión, se los emplea primordialmente en operaciones de ensamblado o empaquetado.
- ❖ **Antropomórfico**. Está formado por tres ejes rotacionales, presenta una gran maniobrabilidad y accesibilidad a zonas con obstáculos, son robots muy rápidos, que permiten trayectorias muy complejas. Sus movimientos tienen similitud con el brazo humano.

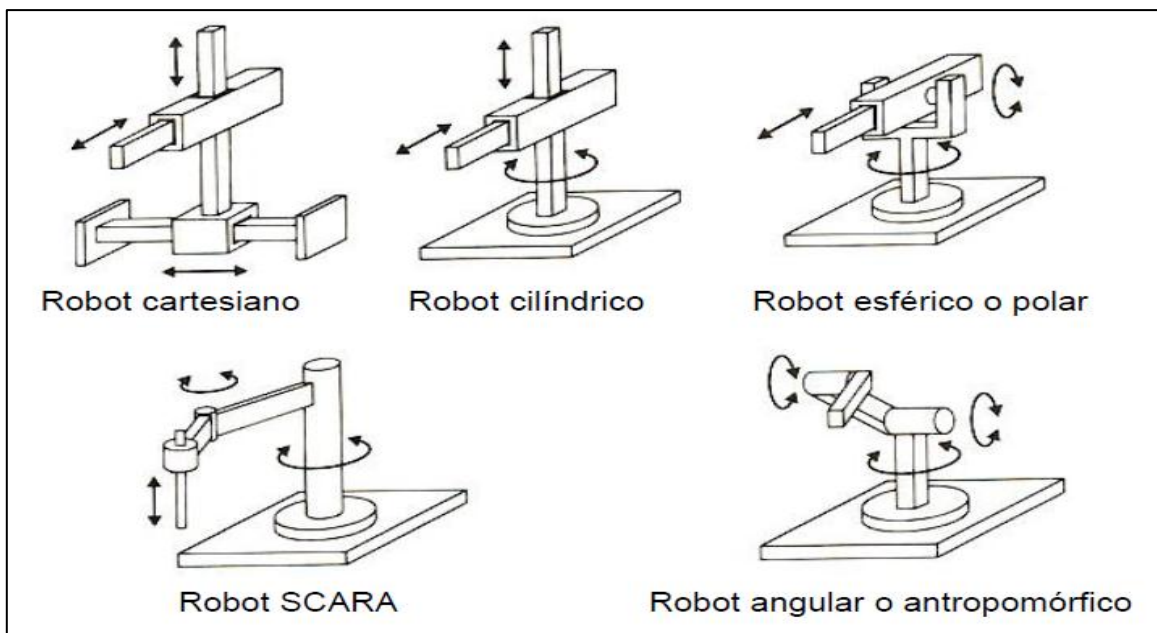


Figura 1-2: Tipos de Robots manipuladores

Fuente: ISA, Ingeniería de Sistemas y Automática. Configuraciones cinemáticas

2.2.10. Cinemática

Quizá uno de los principales problemas en cuanto a robots se relaciona con la posición del robot y su orientación. La cinemática es una parte de la Física que estudia el movimiento de los cuerpos sin considerar su causa. Cuando hablamos de robots la cinemática facilita la posición y orientación del efector final o extremo final del robot, con los valores de las variables articulares como si se tratase de una función de tiempo.

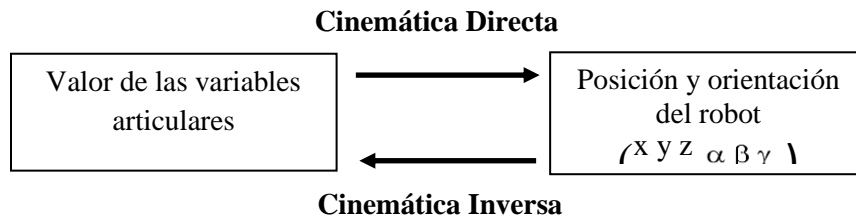


Figura 2-2: Diagra de relación entre cinemática directa e inversa

Fuente: (Barrientos, Peñín, & Balaguer, 1995)

La cinemática directa estudia la posición y orientación de efector final como consecuencia de asignar un valor determinado a cada una de las variables articulares y de resolver la cadena cinmeática con base en tales valores. La cinemática inversa se dedica a encontrar los valores que deben ser asignados a las variabes articulares para posicionar el efector final de modo que su matriz de transformación T_n coincida con una matriz de transformación homogénea (MTH) (Gómez, 2005).

2.2.11. Algoritmos y movimiento

Todo ser humano en su vida cotidiana emplea algoritmos para resolver problemas propios o realizar tareas encomendadas. Varios autores coinciden en que los algoritmos señalan una lista de instrucciones para resolver un cálculo o un problema abstracto, es decir, que un número finito de pasos convierten los datos de un problema (entrada) en una solución (salida).

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. (REA, 2000). Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

No existe restricción alguna sobre el tipo de problema que se pretenda resolver. Sin embargo un algoritmo debe poseer las siguientes características:

- ❖ **Tiempo secuencial.** Hace referencia a que un algoritmo funciona en tiempo discretizado denominado también paso a paso, definiendo así una secuencia de estados

"computacionales" por cada entrada válida (datos que se le proveen al algoritmo antes de comenzar).

- ❖ **Estado abstracto.** Cada estado computacional puede ser descrito formalmente utilizando una estructura de primer orden y cada algoritmo es independiente de su implementación, de manera que en un algoritmo las estructuras de primer orden son invariantes bajo isomorfismo.
- ❖ **Exploración acotada.** La transición de un estado al siguiente queda determinada por una descripción fija y finita (cantidad fija y limitada de términos del estado actual).

Según lo expuesto, un algoritmo es cualquier cosa que funcione paso a paso, donde cada paso se pueda describir con precisión, considerando un límite fijo en la cantidad de datos que se pueden leer/escribir en un solo paso. Dentro de esta definición se incluyen tanto los algoritmos prácticos como los que funcionan en teoría.

2.2.12. Denavit – Hartenberg (D-H)

Denavit – Hartenberg es un método que aplica a las propiedades de las matrices de transformación homogénea (MTH), para representar las relaciones de traslación y rotación entre elementos adyacentes de un robot. En 1995 Denavit y Hartenberg propusieron un algoritmo para establecer de manera sistemática un sistema de coordenadas (S_i) ligado a cada eslabón (i) de una cadena articulada, esto permite pasar de un eslabón al siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón. (Barrientos, Peñín, & Balaguer, 1995)

Después de la aplicación del algoritmo D-H se obtiene la matriz D-H, la cual es introducida en la matriz homogénea (MH). La MH lee la matriz DH fijándose en el parámetro S_i para determinar a que tipo de articulación corresponde para luego identificar la variable articular e ignorar el valor correspondiente a dicha matriz, esto se debe a que MH lee los valores articulares de otra matriz, que debe ser especificada por el usuario, llamada (q). (Gómez, 2005).

2.2.13. Simulación

Thomas Goldsmith en el año 1976, definió la simulación como una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos períodos.

Por su parte Shannon 1976, manifestó que la simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias, dentro de los límites impuestos por un cierto criterio o un conjunto de ellos, para el funcionamiento del sistema.

a. Etapas para realizar un estudio de simulación

- ❖ **Definición del sistema.** Consiste en estudiar el contexto del problema, identificar los objetivos del proyecto, especificar los índices de medición de la efectividad del sistema, establecer los objetivos específicos del modelamiento y definir el sistema en el que se va a modelar la simulación.
- ❖ **Formulación del modelo.** Una vez definidos con exactitud los resultados que se esperan obtener del estudio, se define y construye el modelo con el cual se obtienen los resultados deseados. En la formulación del modelo es necesario definir todas las variables que forman parte de él, sus relaciones lógicas y los diagramas de flujo que describan en forma completa el modelo.
- ❖ **Colección de datos.** Es importante que se definan con claridad y exactitud los datos que el modelo va a requerir para producir los resultados deseados.
- ❖ **Implementación del modelo en la computadora.** Con el modelo definido, el siguiente paso es decidir si se utiliza algún lenguaje o se utiliza algún paquete para procesarlo en la computadora y obtener los resultados deseados.
- ❖ **Verificación.** El proceso de verificación consiste en comprobar que el modelo simulado cumple con los requisitos de diseño para los que se elaboró. Se trata de evaluar que el modelo se comporte de acuerdo al diseño del modelo.

- ❖ **Validación del sistema.** A través de esta etapa se valoran las diferencias entre el funcionamiento del simulador y el sistema real que se trata de simular.
- ❖ **Experimentación.** La experimentación con el modelo se realiza después que este haya sido validado. La experimentación consiste en comprobar los datos generados como deseados y en realizar un análisis de sensibilidad de los índices requeridos.
- ❖ **Interpretación.** En esta etapa del estudio, se interpretan los resultados que arroja la simulación y con esta base se toma una decisión. Es obvio que los resultados que se obtienen de un estudio de simulación permite soportar decisiones del tipo semiestructurado.
- ❖ **Documentación.** Dos tipos de documentación son requeridos para hacer un mejor uso del modelo de simulación. La primera se refiere a la documentación del tipo técnico y la segunda se refiere al manual del usuario, con el cual se facilita la interacción y el uso del modelo desarrollado.

b. Modelos de simulación

La experimentación puede ser un trabajo de campo o de laboratorio. El modelo de método usado para la simulación puede ser teórico, conceptual o sistémico. (Arechavaleta, 2014).

- ❖ **Modelo teórico.** Debe contener los elementos que se precisen para la simulación y ser capaz de determinar la normalidad en términos de probabilidad de las series generadas. Es el modelo principal de toda investigación científica, gracias a ello se pueden definir o concluir la hipótesis, las predicciones, etc.
- ❖ **Modelo conceptual.** Desea establecer por un cuestionario y con trabajo de campo, la importancia de la discriminación o rechazo en una colectividad a través de una escala de actitud. Gran parte de las simulaciones se realizan con modelos conceptuales.
- ❖ **Modelo sistémico.** El modelo sistémico se construye utilizando como metodología la dinámica de sistemas, en una representación total. Este método, es sumamente abstracto, y

no se limita a la descripción del sistema, sino que debe incluir en la simulación las entradas y salidas de energía y los procesos de retroalimentación.

Un modelo se utiliza como ayuda para el pensamiento al organizar y clasificar conceptos confusos e inconsistentes. Al realizar un análisis de sistemas se crea un modelo del sistema que muestre las entidades, las interrelaciones, etc. La adecuada construcción de un modelo ayuda a organizar, evaluar, examinar la validez de pensamientos.

2.3. El sistema robótico kawasaki RS 003N

El robot KAWASAKI RS 003N, es un Robot industrial de 6GL, con un diseño compacto ideal para una amplia gama de aplicaciones (manipulación, ensamble, manejo de materiales, etc.), soporta una carga máxima de 3kg, posee un radio de acción de 620 mm.



Figura 3-2: Robot Kawasaki RS003N

Fuente: (Kawasaki, 2016) <https://robotics.kawasaki.com/en1/products/robots/small-medium-payloads/RS003N/>

2.3.1. Características del Robot Kawasaki RS003N

- ❖ **Alto rendimiento.** Los motores de altas revoluciones, la aceleración y la operación de alta velocidad lo convierten en líder de la industria robótica. La tasa de aceleración se ajusta automáticamente para adaptarse a la carga útil y la postura robot para entregar óptimo rendimiento y tiempos de ciclo más cortos.

- ❖ **Amplio rango de trabajo.** El ángulo de giro de cada eje, proporciona un rango de movimiento extendido, lo cual se traduce en una mayor área de trabajo, un mínimo de espacio muerto y mayor flexibilidad.
- ❖ **Construcción liviana.** El brazo está constituido por materiales conformados por una aleación de aluminio, esta técnica de construcción hace que el brazo sea liviano pero rígido, lo que permite la gran velocidad y exactitud del sistema.
- ❖ **Protección del medio ambiente.** Incorpora una construcción de doble sello en todos los ejes y las conexiones eléctricas son resistentes al agua.
- ❖ **Ahorro de espacio.** El diseño del brazo delgado requiere muy poco espacio en el suelo.
- ❖ **Fácil mantenimiento.** El diseño de Kawasaki requiere muy poco, cada articulación esta sujeta por rodamiento de bolas de alto grado de precisión, las que no necesitaran mantenimiento durante la vida útil del brazo. Además, es líder en la industria con el promedio mínimo de fallos.

2.3.2. *Especificaciones técnicas del Robot Kawasaki RS003N*

En la tabla siguiente se muestran las especificaciones técnicas del Robot Kawasaki RS003N.

Tabla 2-2: Especificaciones técnicas del Robot Kawasaki RS003N

ESPECIFICACIONES RS003N			
Grados de libertad	6 ejes		
Carga útil	3 kg		
Horizontal Alcance	620 mm		
Repetibilidad	± 0.02 mm		
Trabajo Envelope (grados) y velocidad máxima (grados / s)	Eje	Rango de movimiento	Velocidad máxima
	JT1	$\pm 160^\circ$	$360^\circ / s$
	JT2	$+ 150^\circ \sim -60^\circ$	$250^\circ / s$
	JT3	$+ 120^\circ -150^\circ \sim$	$225^\circ / s$
	JT4	$\pm 360^\circ$	$540^\circ / s$
	JT5	$\pm 135^\circ$	$225^\circ / s$
	JT6	$\pm 360^\circ$	$540^\circ / s$
Muñeca de carga Capacidad	Eje	Par máximo	Momento de inercia
	JT4	$5.8 \text{ N} \cdot \text{m}$	$0,12 \text{ kg} \cdot \text{m}^2$
	JT5	$5.8 \text{ N} \cdot \text{m}$	$0,12 \text{ kg} \cdot \text{m}^2$
	JT6	$2.9 \text{ N} \cdot \text{m}$	$0,03 \text{ kg} \cdot \text{m}^2$
Vertical Alcance	20 kg (excluidas las Opciones)		
Color del cuerpo	Kawasaki Estándar		
Instalación	Suelo, techo		
Protección Clasificación	IP54 * Equivalente		
Controlador	América	E76	
	Europa	E70	
	Japón y Asia	E73	

Fuente: (Kawasaki, 2016) <https://robotics.kawasaki.com/en/products/robots/small-medium-payloads/RS003N>

2.3.3. Generalidades del sistema

Procesador
Tarjeta controladora
Drivers x/y
Sensores
Interfaz Humano Máquina

2.3.4. Módulo de Control

Direcciona las acciones que se desea que realice el robot, para lo cual deben establecerse ciertos criterios u objetivos del control.

Los módulos de control determinan las siguientes características del robot:

❖ **Adaptabilidad.** Capacidad de ejecutar tareas en ambientes variables sin asistencia.

- ❖ **Polivalencia.** Capacidad de ejecutar diferentes clases de acciones o tareas.
- ❖ **Grado de automatismo.** Capacidad de ejecutar tareas sin recurrir a un operador humano.

Los tipos de tareas para las cuales los robots son diseñados, las características de estos dispositivos y el nivel de desempeño asociado colocan demandas sobre el sistema de control, que llevan configuraciones más complejas que aquellas utilizadas usualmente en otros tipos de máquinas.

2.3.5. Actuadores

Realizan las acciones programadas bajo la supervisión de la unidad de control. Estos dispositivos pueden ir desde los casos más elementales, como accionadores hidráulicos, neumáticos o electromecánicos hasta máquinas más complejas como manipuladores.

La coordinación entre ambos componentes mediante el intercambio de información es lo que permite conseguir la realización correcta de las tareas a realizar.

2.4. Software empleado para la simulación y control y de un robot manipulador

2.4.1. SolidWorks

SolidWorks, es una herramienta de modelado CAD, que permite el desarrollo de piezas, ensambles y planos de gran variedad de sistemas y mecanismos, además permite realizar simulaciones y análisis básicos CAE.

El término CAD (Computer Aided Design o Diseño Asistido por Ordenador) hace referencia a una herramienta software que, mediante el uso del ordenador, permite crear, modificar, analizar y optimizar planos y modelos en dos y tres dimensiones, y manipular de una manera fácil elementos geométricos sencillos. Se trata de herramientas que van más allá del concepto de “dibujo” o representación gráfica. (Bonilla, 2003).

Tabla 3-2: Requerimientos para el funcionamiento de Solidworks

Sistemas operativos	SOLIDWORKS	SOLIDWORKS	SOLIDWORKS
----------------------------	-------------------	-------------------	-------------------

	2015 (EDU 2015-2016)	2016 (EDU 2016-2017)	2017 (EDU 2017-2018)
Windows 10, 64 bits	✓	✓	✓
Windows 8.1, 64 bits	✓	✓	✓
Windows 8.0, 64 bits	✓	✗	✗
Windows 7 SP1, 64 bit	✓	✓	✓
Hardware			
RAM	Se recomiendan 8 GB o más		
Espacio en disco	5 GB o más		
Tarjeta de vídeo	BOXX, Dell, Fujitsu, HP, Lenovo, MSI, Microsoft, NEC, NVIDIA, Panaonic, Toshiba, Virtual Desktop, Wacom.		
Procesador	Intel o AMD con compatibilidad con SSE2. Se requiere sistema operativo de 64 bits		
Medio de instalación	Unidad de DVD o conexión a Internet de banda ancha		
Productos de Microsoft			
Internet Explorer	IE 9, 10, 11	IE 10, 11	IE 11
Excel y Word	2010, 2013	2010, 2013, 2016 (2016 SP3)	2010, 2013, 2016
Otros			
ANTIVIRUS	Avast, AVG, CA eTrust, ESET NOD32, eScan, iolo, Kaspersky AV, McAfee, Norton, Panda, Sophos, Symantec Endpoint Protection, Trend Micro OfficeSca.		
Red	Redes de Microsoft Windows. Las redes Novell y los dispositivos de almacenamiento de red de marca distinta a Windows no son compatibles.		
Entornos virtuales	Entornos virtuales compatibles		

Fuente: <https://www.solidworks.es/sw/support/SystemRequirements.html>

2.4.2. Matlab

Es un lenguaje de alta performance utilizado primordialmente para cálculo técnico. El mismo software integra cálculo, visualización y programación en un entorno de fácil utilización en donde los problemas y las soluciones son expresados en una notación matemática familiar. Los usos más típicos incluyen cálculos matemáticos, desarrollo de algoritmos, modelado, simulación y prototipos, gráficas científicas e ingenieriles. (Matlab, 2016).

En definitiva, Matlab permite realizar cálculos técnicos científicos y de propósito general. Integra operaciones de cálculo, visualización y programación, además emplea una notación matemática clásica para la interacción con el usuario. Dentro de sus principales usos y aplicaciones podemos mencionar a:

- ❖ Cálculo y Matemáticas.
- ❖ Desarrollo de algoritmos.
- ❖ Adquisición, análisis y procesamiento de datos.
- ❖ Modelado y simulación.
- ❖ Gráfica científica y de ingeniería.
- ❖ Desarrollo de aplicaciones.

2.4.3. Componentes de Matlab

Integra cuatro aspectos fundamentales:

- 1) **Entorno de desarrollo.** Gama de utilidades que admiten el uso de las funciones Matlab y ficheros en general.
- 2) **Librería de funciones matemáticas.** Conjunto de algoritmos de cálculo, comprendiendo las funciones más elementales como la suma, senos y cosenos o la aritmética compleja, hasta funciones más sofisticadas como la inversión de matrices, el cálculo de autovalores, funciones de Bessel y transformadas rápidas de Fourier.
- 3) **Gráficos.** Gama de utilidades destinadas a visualizar vectores y matrices en forma de gráficos, destacando la visualización tridimensional con opciones de iluminación y sombreado, a demás de la posibilidad de crear animaciones.
- 4) **Interfaz de aplicación.** Librería que permite escribir programas ejecutables independientes en C y otros lenguajes, accediendo, mediante DLLs, a las utilidades de cálculo matricial de Matlab.

Tabla 4-2: Requisitos adicionales de Matlab

Requisitos adicionales de Matlab y Simulink	Windows (64-bit)	Mac (64-bit)	Linux (64-bit)
DO Kit de calificación	X	X	X

Codificador incorporado	X	X	X
Diseño de filtro HDL Coder	X	X	X
Diseñador de puntos fijos	X	X	X
Codificador HDL	X	X	X
Verificador de HDL	X	X	X
Kit de certificación IEC	X	X	X
MATLAB	X	X	X
Codificador MATLAB	X	X	X
Compilador de MATLAB	X	X	X
MATLAB Compiler SDK	X	X	X
Servidor de computación distribuida MATLAB	X	X	X
Servidor de Producción MATLAB	X	X	X
Generador de informes MATLAB	X	X	X
Modelo de cuadro de herramientas de control predictivo	X	X	X
Cuadro de herramientas de ecuaciones diferenciales parciales	X	X	X
Polyspace Bug Finder	X	X	X
Cliente de Polyspace para Ada	X	X	X
Polyspace Code Prover	X	X	X
Servidor de Polyspace para Ada	X	X	X
Blockset del grupo motopropulsor	X	X	X
Simbiología	X	X	X
SimEventos	X	X	X
SimRF	X	X	X
Simscape	X	X	X
Simscape Driveline	X	X	X
Simscape Electrónica	X	X	X
Fluidos Simscape	X	X	X
Simscape Multibody	X	X	X
Sistemas de energía de Simscape	X	X	X
Simulink	X	X	X
Animación 3D de Simulink	X	X	X
Inspector de código de Simulink	X	X	X
Simulink Coder	X	X	X
Simulink Control Design	X	X	X
Optimización del diseño de Simulink	X	X	X

Simulink Desktop en tiempo real	X	X	X
Verificador del diseño de Simulink	X	X	X
Simulink PLC Coder	X	X	X
Simulink en tiempo real	X	X	X
Generador de informes Simulink	X	X	X
Prueba de Simulink	X	X	X
Verificación y validación de Simulink	X	X	X
Enlace de hoja de cálculo	X	X	X
Flujo de estado	X	X	X
Herramientas de aprendizaje	X	X	X

Fuente: <https://es.mathworks.com/products/availability.html#ML>

2.4.4. Simulink

Constituye una herramienta muy útil en cuanto a la simulación de sistemas dinámicos. Es una aplicación que permite construir y simular modelos de sistemas físicos y sistemas de control a través de diagramas de bloques en un entorno de trabajo gráfico. Dispone de una serie de utilidades que facilitan la visualización, análisis y guardado de los resultados de simulación.

2.4.5. Bibliotecas fundamentales de SimMechanics

SimMechanics, es una herramienta de Simulink que permite la simulación de sistemas mecánicos mediante la utilización de bloques que representa las piezas del sistema, articulaciones, restricciones y elementos actuantes, también permite la importación de modelos CAD 3D ya desarrollados, admiten generar un control y simular permitiendo incluso ver la dinámica del sistema. (MathWorks, 2016).

Dentro de SimMechanics se encuentran varias bibliotecas que permiten modelar sistemas mecánicos por medio de bloques y simular dichos sistemas para conocer su comportamiento en ambientes idóneos. Las bibliotecas de SimMechanics se muestran en la figura 4, siendo las principales: bodies, joints y sensors & actuators.

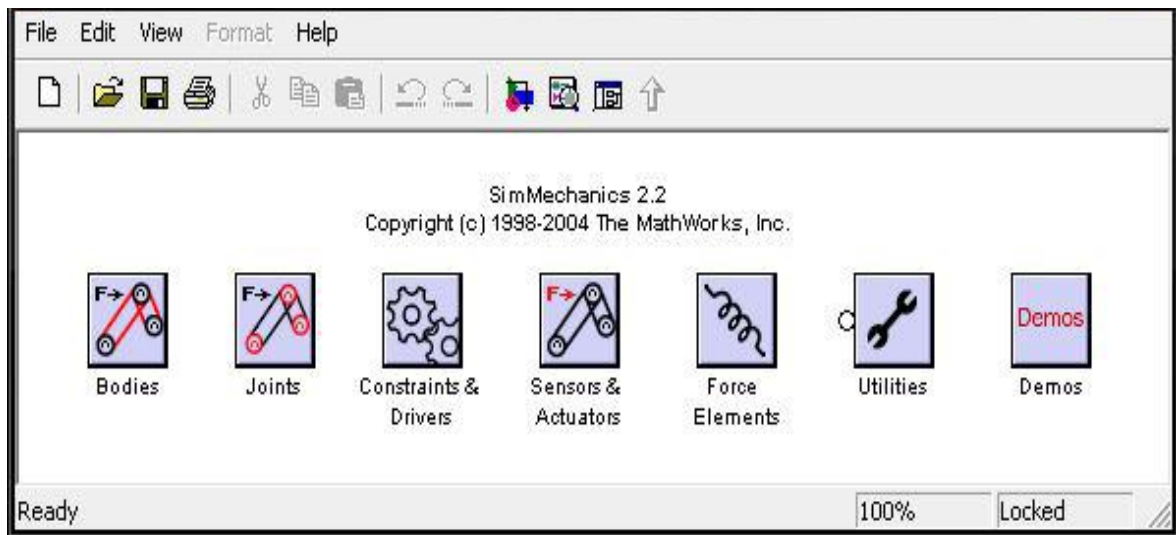


Figura 4-2: Bibliotecas de SimMechanics

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

a. Bodies o cuerpos

Esta biblioteca integra los elementos más importantes para modelar un sistema mecánico. Contiene tres bloques: body (cuerpo), ground (tierra) y machine environment (ambiente).

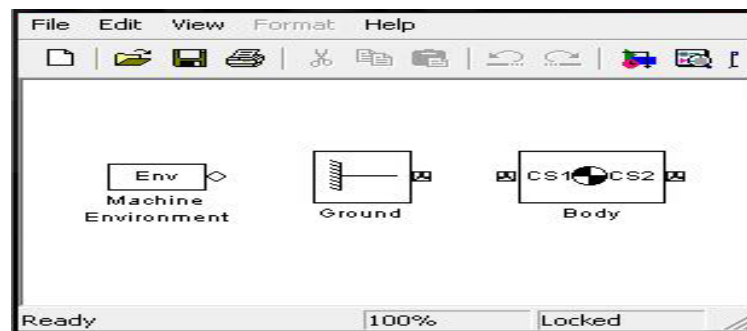


Figura 5-2: Bibioteca Bodies

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

b. Joints o junturas

Componen los diferentes tipos de articulaciones que unen los cuerpos del sistema y determinan los grados de libertad con los que cuenta el sistema.

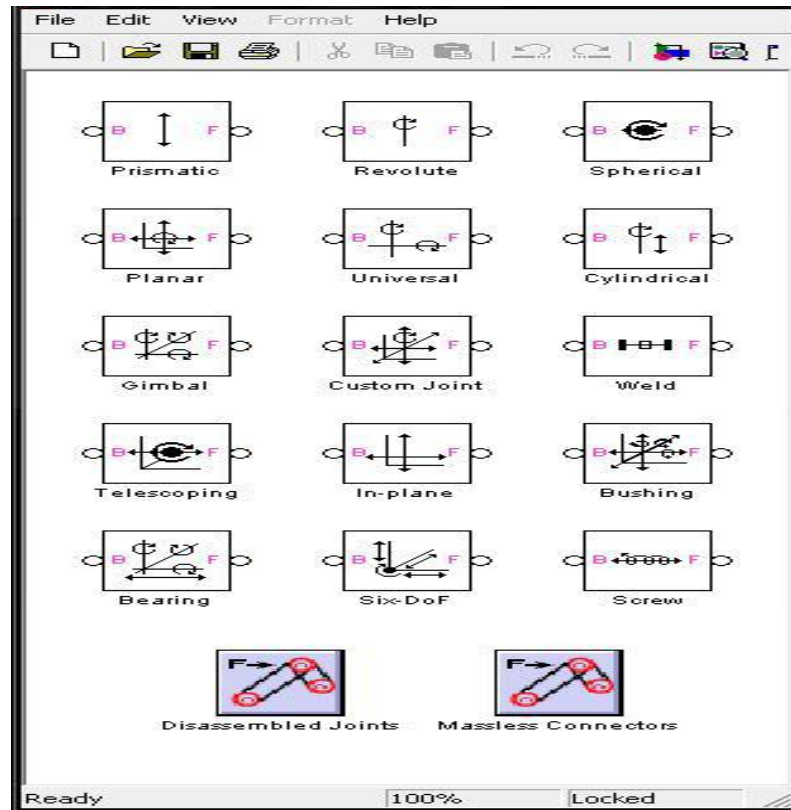


Figura 6-2: Biblioteca Joints

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

c. Sensors & Actuators

Contiene los bloques que unen y permiten la interacción entre bloques de SimMechanics y Simulink.

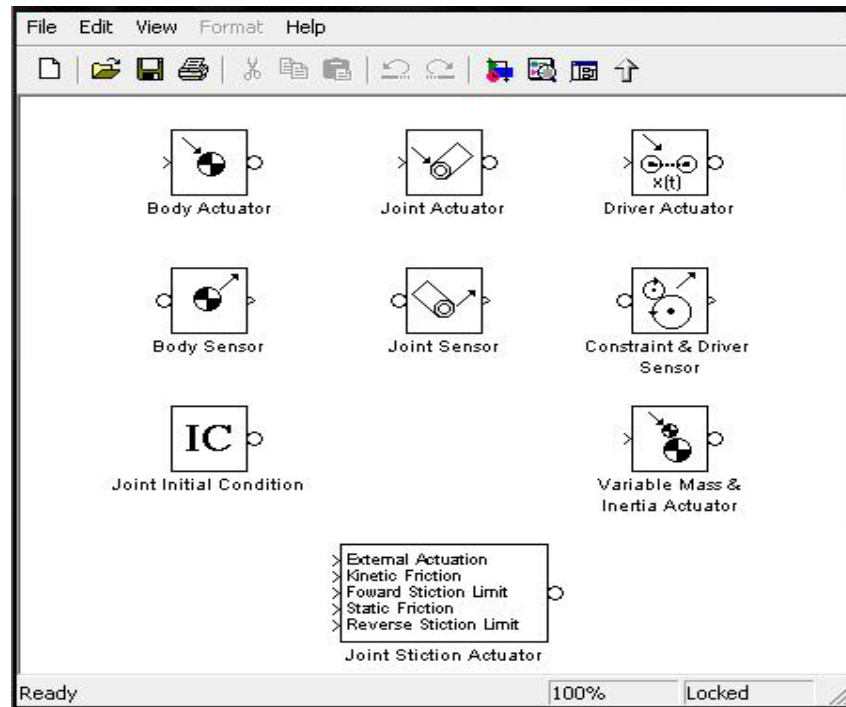


Figura 7-2: Bibioteca Sensors & Actuators

Fuente: Captura de pantalla Bibioteca Sensors & Actuators

Elaborado por: Irma Villa 2017

CAPÍTULO III

3. DISEÑO DE INVESTIGACIÓN

3.1. Modalidad de la Investigación

La presente investigación tiene relación con elementos de orden teórico-conceptuales y con perspectivas analíticas-interpretativas, a partir de lo cual se recopilan, ordenan, e interpretan los datos y experiencias obtenidos durante el proceso investigativo, para luego convertirlos en aplicaciones tecnológicas que permitan dar movimiento al Robot Kawasaki RS003N.

3.2. Tipos de Investigación

Respecto a los tipos de investigación, el presente estudio se caracteriza por ser:

- ❖ **Documental.** Se han analizado libros, técnicas, revistas y artículos científicos, para sustentar teóricamente la investigación.
- ❖ **Experimental.** La investigación se ha desarrollado caracterizando y poniendo en práctica, diversas aplicaciones tecnológicas a fin de identificar las más adecuadas para realizar el control de los movimientos del Robot Kawasaki RS003N.

3.3. Métodos de investigación.

Dentro de la metodología aplicada se ha considerado el método inductivo-deductivo, ya que se han revisado las teorías y caracterizado las tecnologías que posteriormente se han puesto en práctica para modelar un algoritmo idóneo que permita controlar el movimiento de Robot Kawasaki RS003N.

3.4. Técnicas e Instrumentos de recolección de datos.

Durante el desarrollo del presente estudio se utilizan como técnicas:

- ❖ **Observación directa y participante.** Ha sido puesta en práctica durante todo el proceso investigativo, ya que la investigadora forma parte activa en la solución de la problemática identificada.
- ❖ **Entrevista.** Se ha aplicado a entendidos en la materia a fin de conocer sus puntos de vista y sus experiencias con las diversas aplicaciones tecnológicas que se pretenden emplear para controlar el movimiento de Robot Kawasaki RS003N.

3.5. Diseño del proyecto

- ❖ Diseño y Modelamiento del Brazo Robótico en 3D utilizando Solidworks y CAD.
- ❖ Importación del Modelo CAD a Simulink.
- ❖ Establecimiento de la estructura mecánica del brazo robótico en Simulink por medio de la herramienta SimMechanics.
- ❖ Desarrollo del algoritmo de control para insertar las trayectorias al manipulador.
- ❖ Realización de pruebas de simulación y control de los movimientos del brazo robótico utilizando Matlab y Simulink.

3.6. Presupuesto

Tabla 1-3: Presupuesto

Cantidad	Descripción	V. Unitario	V. Total
1	Capacitación sobre e funcionamiento de Solidworks	320.00	320.00
1	Capacitación sobre e funcionamiento de Matlab	320.00	320.00
1	Capacitación sobre e funcionamiento de Simulink - SimMechanics	320.00	320.00
Total		960.00	960.00

Fuente: Autofinanciamiento

Elaborado por: Irma Villa 2017

CAPÍTULO IV

4. SIMULACIÓN Y CONTROL DEL BRAZO ROBÓTICO KAWASAKI RS 003N

4.1. Análisis del Robot Kawasaki RS003N

Como ya se ha mencionado el Kawasaki RS 003N, es un robot industrial que posee 6 grados de libertad (GL) y un diseño compacto ideal para una amplia gama de aplicaciones como ensamble, manejo de materiales, entre otros. Incluye un dispositivo de memoria...

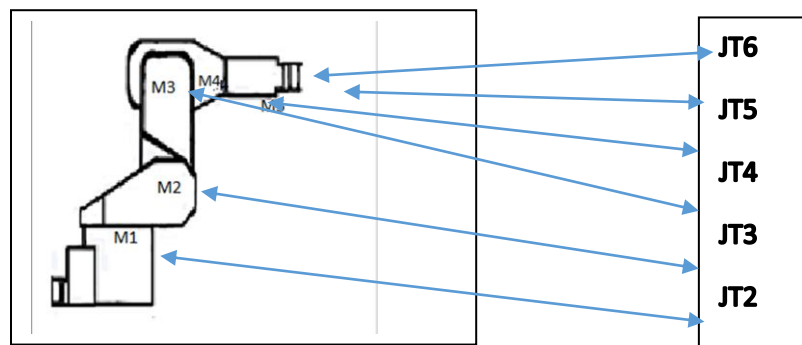


Figura 1-4: Esquema con la ubicación de los ejes del Robot Kawasaki RS003N

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.1.1. Mecanismos del robot

Sistemas reductores y engranajes

El sistema reductor del Robot Kawasaki RS003N permite mover sus eslabones, para lo cual una vez definida la orden, disminuye la velocidad del actuador y transmite el movimiento de un punto físico a otro incrementando la fuerza del actuador, con lo cual es posible movilizar un eslabón de mayor tamaño o a su vez una serie de eslabones interconectados.

El sistema del actuador del Kawasaki RS003N, está compuesto por un actuador rotación

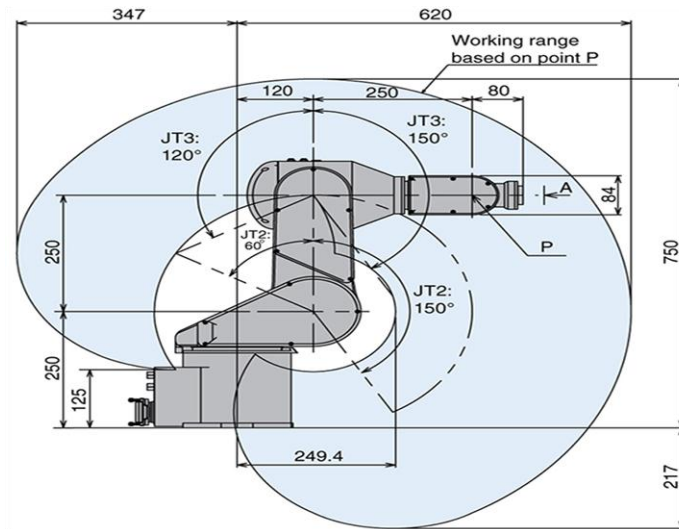


Figura 2-4: Rango y dimensiones del Robot Kawasaki RS 003N

Fuente: <https://robotics.kawasaki.com/en/1/products/robots/small-medium-payloads/RS003N/>

De acuerdo con las especificaciones técnicas del robot, el eje (JT1) tiene un giro de 320° , el eje (JT2) tiene un ángulo de 150° , el eje (JT3) tiene 150° de giro, el eje (JT4) posee 360° de movimiento, el eje (JT5) cuenta con 360° de giro y finalmente el eje (JT6) posee un ángulo de apertura de 60° .

Cumple con las características de robot establecidas por la Organización Internacional de Estándares (ISO), como “un manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”.

Tabla 1-4: Clasificación de los robots según la AFRI

Tipo	Descripción
A	Manipulador con control manual o telemando.
B	Manipulador automático con ciclos pre-ajustados; regulación mediante fines de carrera o topes, control por PLC; accionamiento neumático, eléctrico o hidráulico.
C	Robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno
D	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de éstos

Fuente: Asociación Francesa de Robótica Industrial (AFRI)

Elaborado por: Irma Villa 2017

De acuerdo con la clasificación de la (AFRI), el Kawasaki RS 003N, es un robot Tipo A, por otro lado, al poseer la facultad de grabar de movimientos, se trataría de un robot de Tipo C, robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno.

4.2. Diagrama del proyecto

El presente proyecto contempla el modelo del robot en 3D en el programa CAD, seguidamente su importación a Simulink, la definición del modelo de control en Matlab y la comprobación del proyecto mediante ejercicios de simulación que pueden ser realizados por el Robot físico Kawasaki.

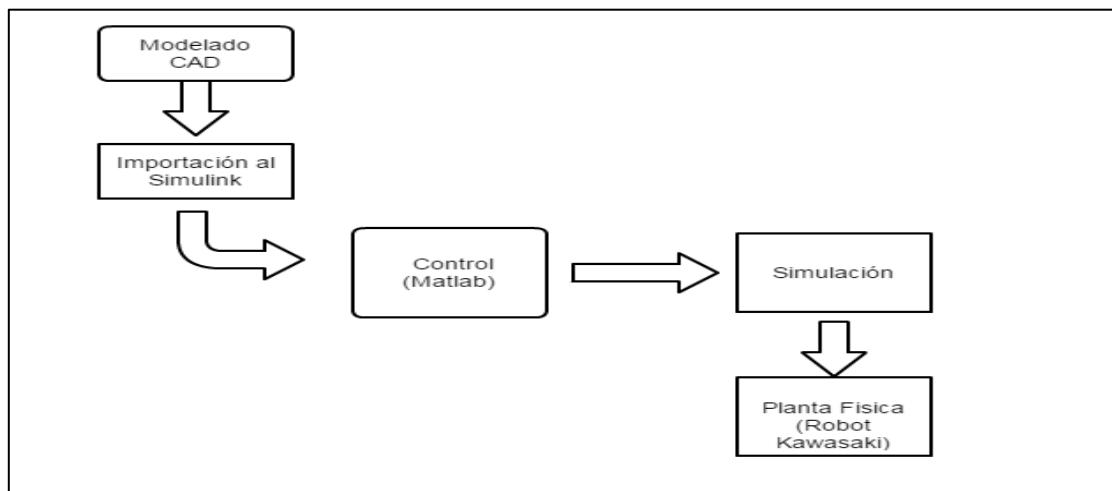


Figura 3-4: Diagrama del Proyecto

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.3. Requerimientos del Sistema

En cuanto a los requisitos mínimos empleados en el proyecto podemos mencionar los siguientes:

- ❖ Procesador Intel Pentium III 800 MHz.
- ❖ Sistema Operativo Windows 8
- ❖ 500 MB de RAM.
- ❖ 500 MB de espacio de disco duro.
- ❖ XGA display (1024x768)
- ❖ 1 puerto USB.

- ❖ Bluetooth.
- ❖ Visual C++
- ❖ Matlab 2016 a
- ❖ Solid Works 2016

En cuanto al software, se ha procedido a instalar las librerías necesarias en el ordenador donde se pretende ejecutar el programa. Dentro del software empleado se encuentran los programas Solidworks y Matlab. Posteriormente se han realizado las comprobaciones previas de las conexiones, así como la asignación de los puertos de los periféricos. Finalmente se ha realizado la ejecución del programa, con lo cual el usuario tiene la capacidad de emitir órdenes para controlar el brazo robótico y visualizar la simulación en la pantalla del ordenador. Para la instalación del software se siguieron los pasos establecidos para cada software.

4.4. Diagrama de Conexión del Modelo

En primera instancia se procede al modelado del brazo robótico en Solidworks, luego se importa dicho modelado a Simulink. Para el control se emplea la librería de Peter Corke, como se analizará luego en la sección 4.8.3.2, y el Algoritmo Denavit – Hartenberg. Finalmente se realizan las respectivas pruebas de funcionamiento.

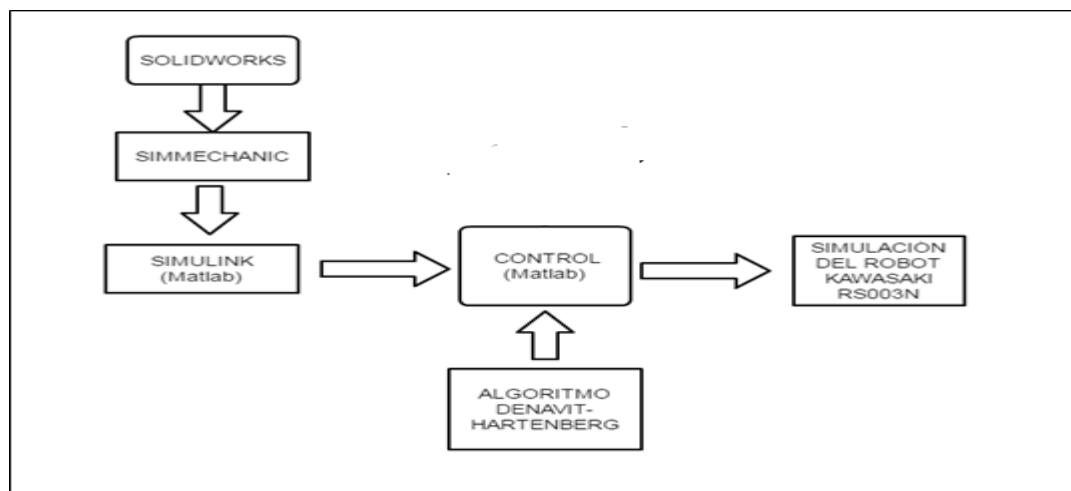


Figura 4-4: Diagrama de Conexión del Modelo

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.5. Modelado del Brazo Robótico utilizando Solidworks de CAD.

Solidworks es un software CAD muy amigable para modelado mecánico de piezas y cuerpos en 3D, el desarrollo del modelado en Solidworks es más sencillo que en SimMechanics, por tal razón se lo ha considerado para el modelamiento de las piezas del Kawasaki RS 003N, considerando las medidas ya optimizadas realizadas por medio de cálculos de Matlab.

Creando un nuevo documento de Solidworks, seleccionando la opción “una representación en 3D de un único componente de diseño”, con lo cual se da origen a una nueva área de trabajo y seleccionando la opción croquis podemos tener acceso a las diferentes formas o figuras base para el diseño de cada una de las piezas. Es necesario conocer las medidas así como los parámetros de las figuras base, de acuerdo con el diseño físico del robot.

En primera instancia se ha utilizado en la base del robot figuras como rectángulos, cuadrados y círculos.

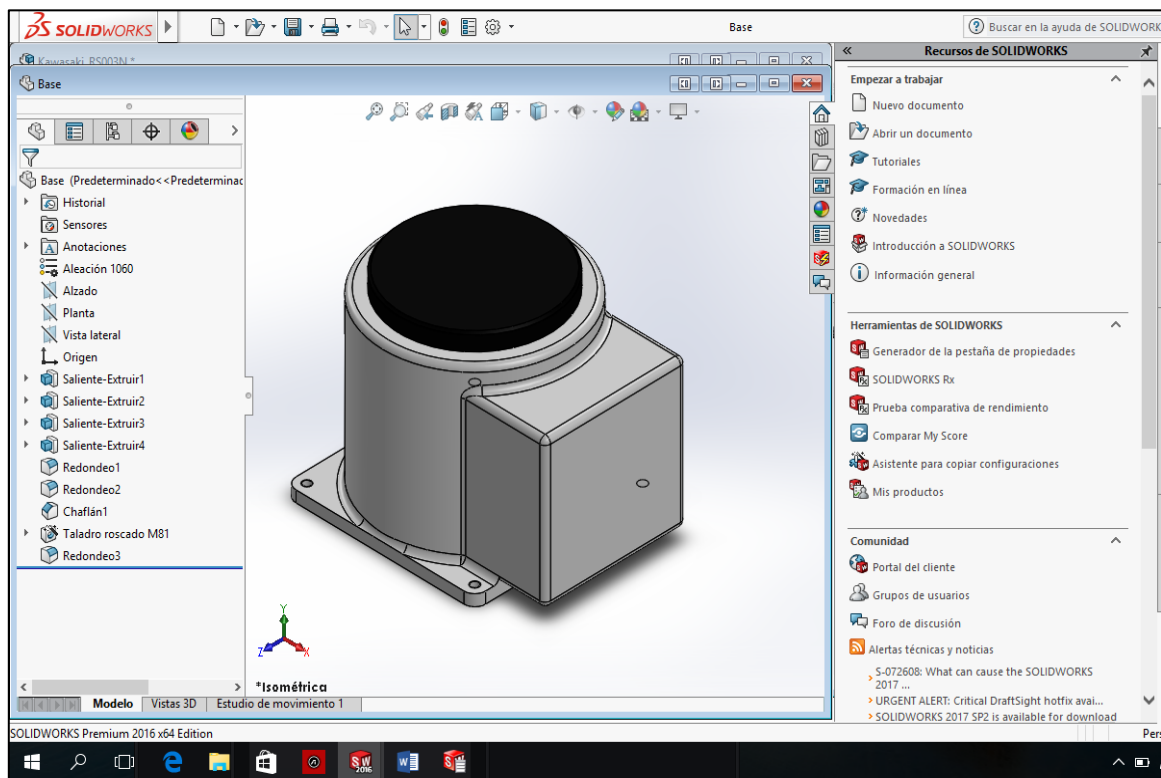


Figura 5-4: Diseño de la base del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Posteriormente se procedió con el diseño de los cuerpos uno, dos, tres, cuatro, cinco y la pinza, para lo cual se tomaron como base figuras bases disponibles en Solidworks, mismas que se acoplaron para dar origen a cada una de las piezas.

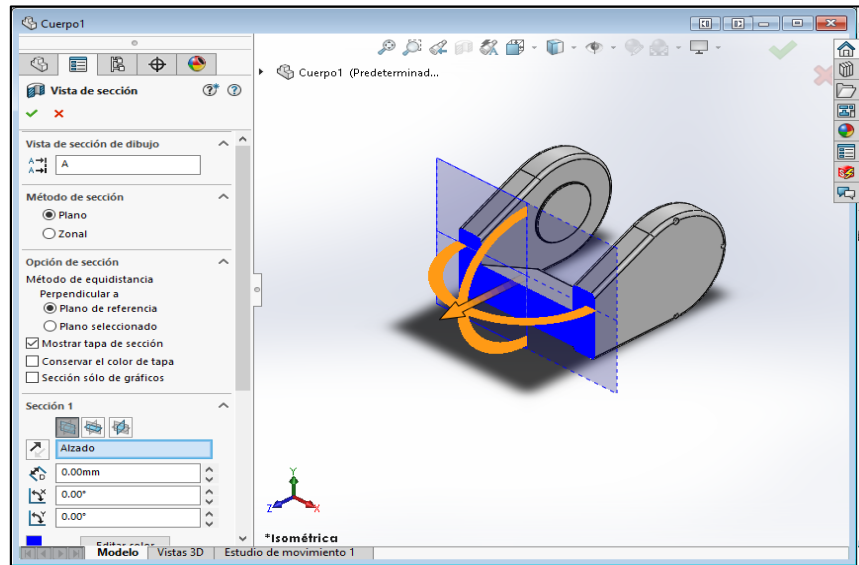


Figura 6-4: Diseño del cuerpo 1 del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

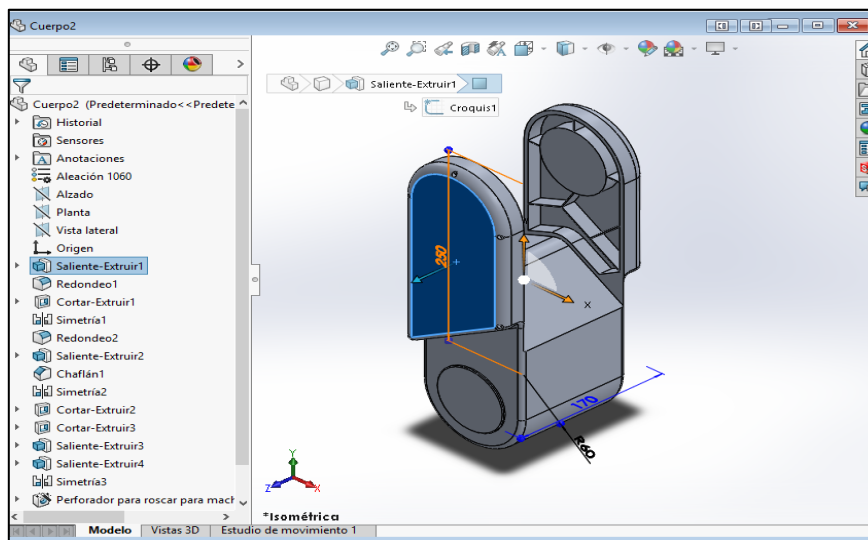


Figura 7-4: Diseño del cuerpo 2 del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

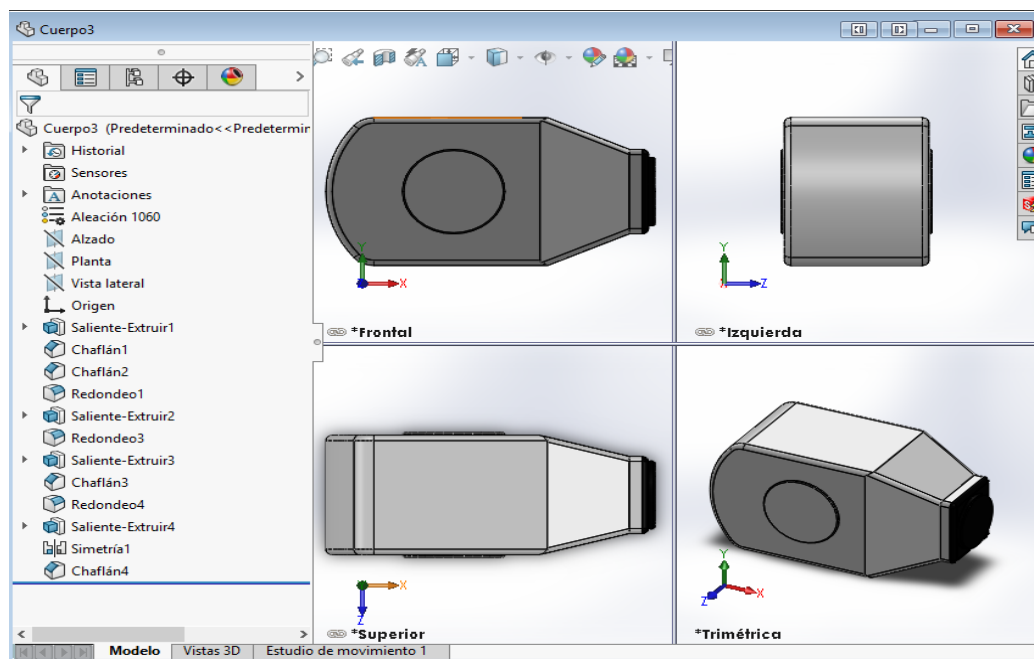


Figura 8-4: Diseño del cuerpo 3 del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

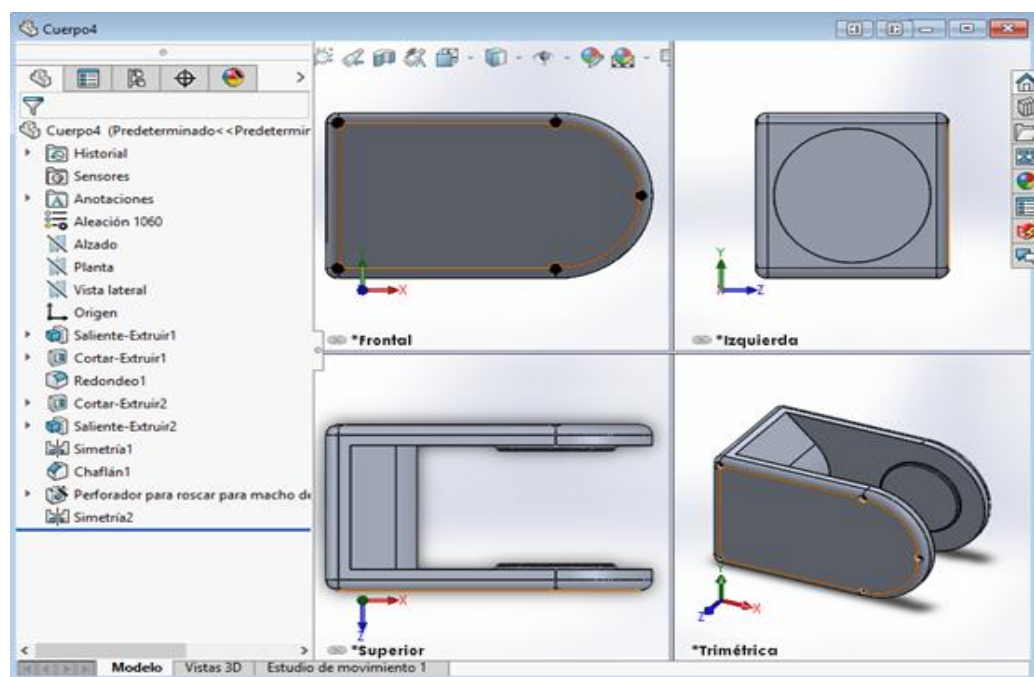


Figura 9-4: Diseño del cuerpo 4 del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

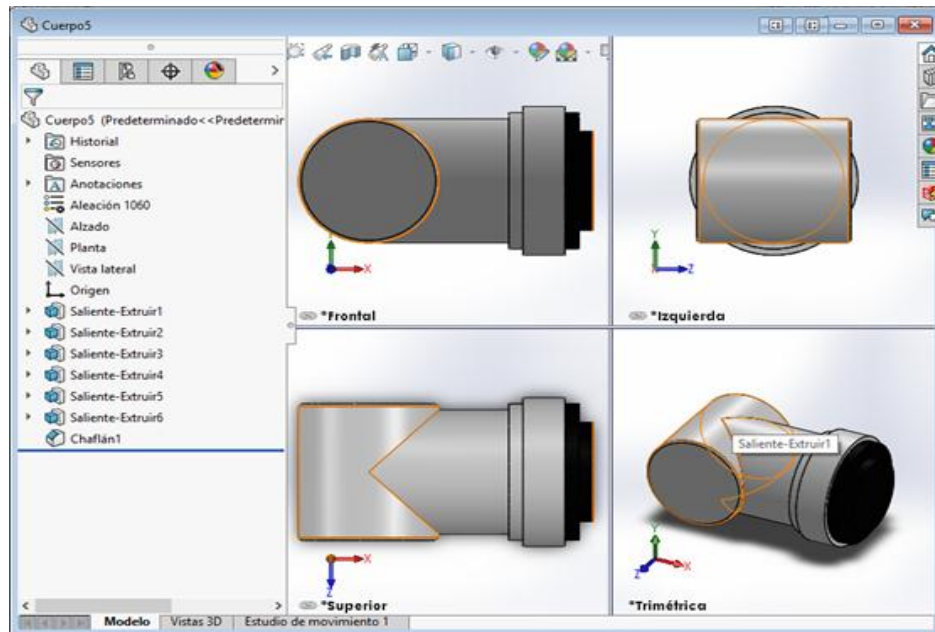


Figura 10-4: Diseño del cuerpo 5 del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

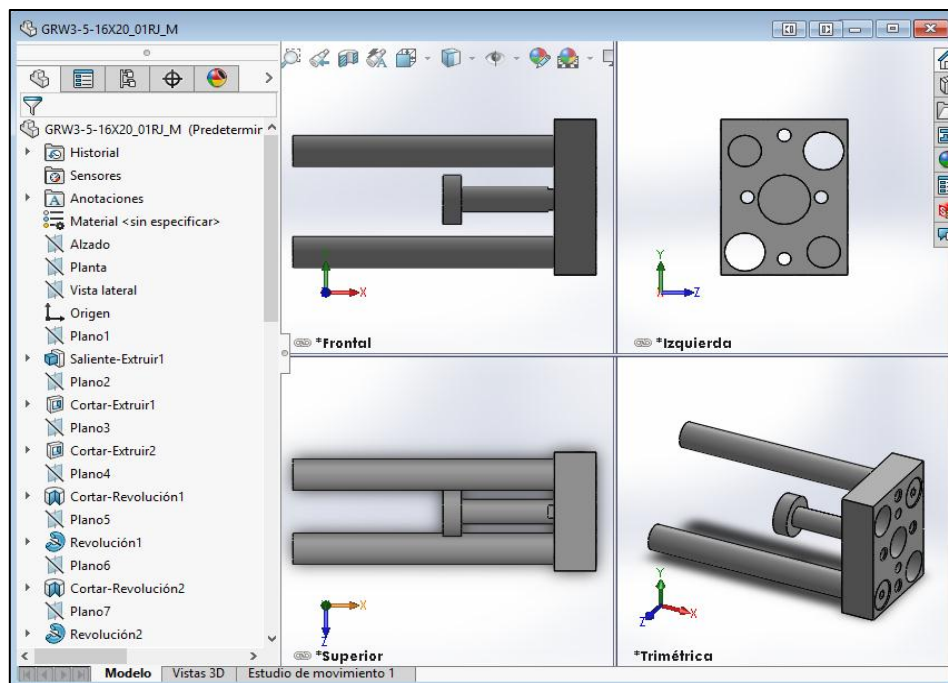


Figura 11-4: Diseño de la mordaza del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

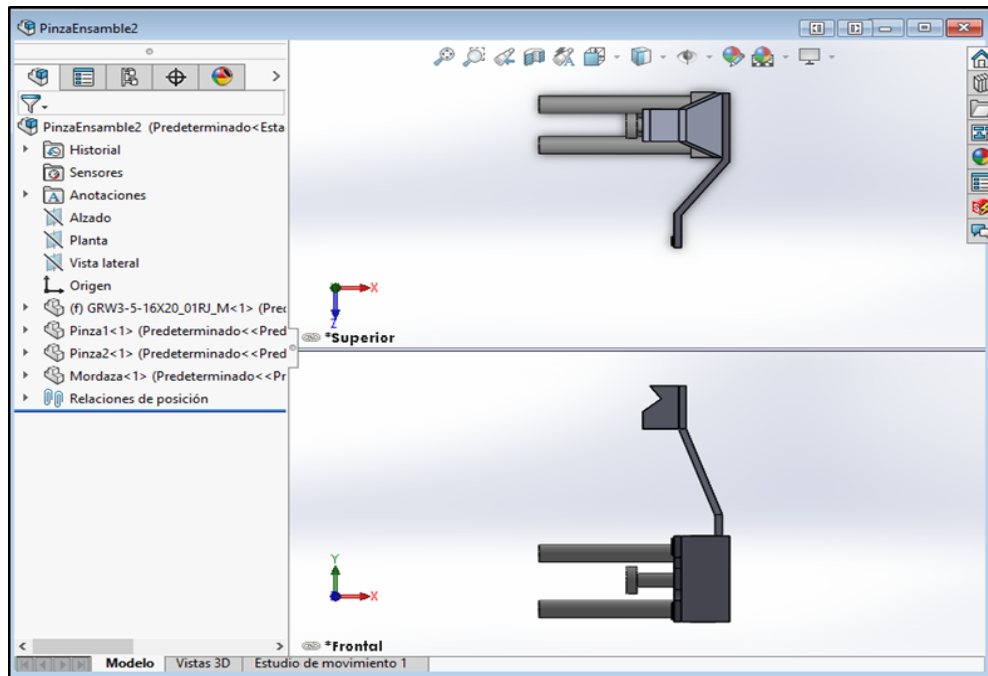


Figura 12-4: Diseño de la pinza del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

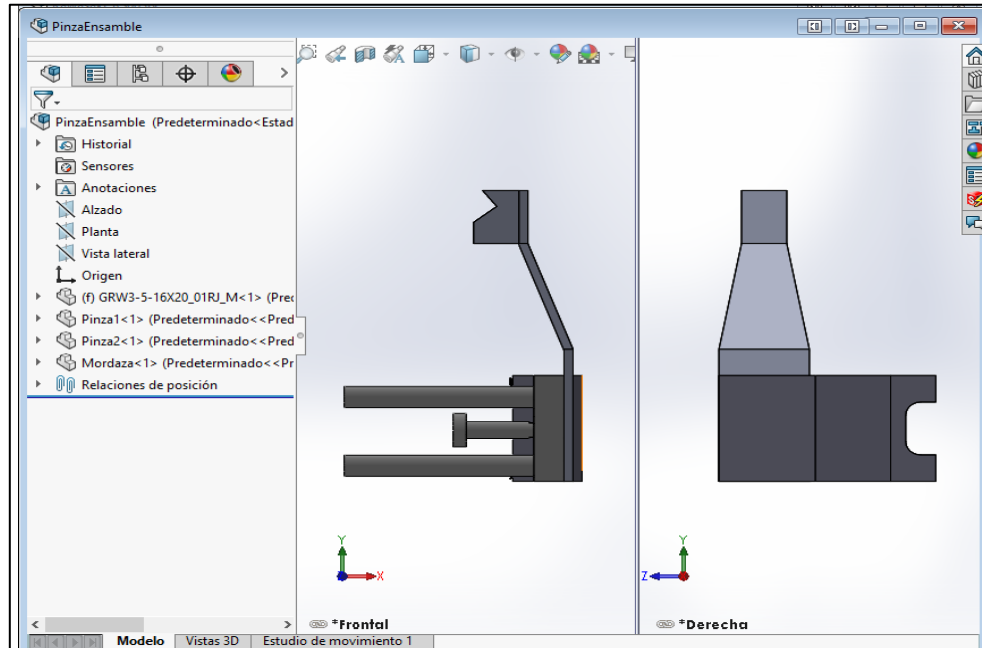


Figura 13-4: Ensamblaje de la pinza en el brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

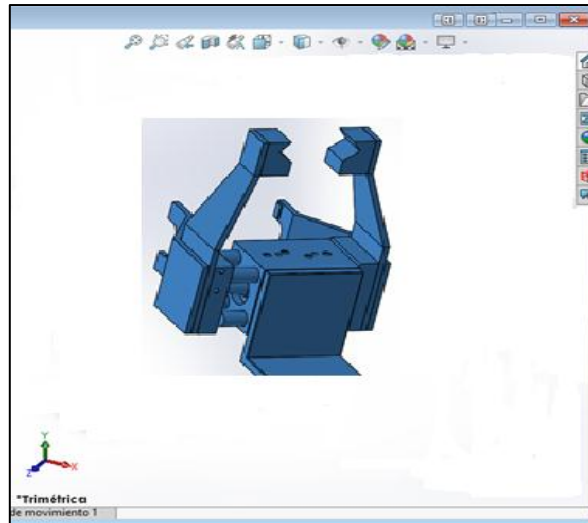


Figura 14-4: Vista en 3D de la pinza ensamblada completa

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

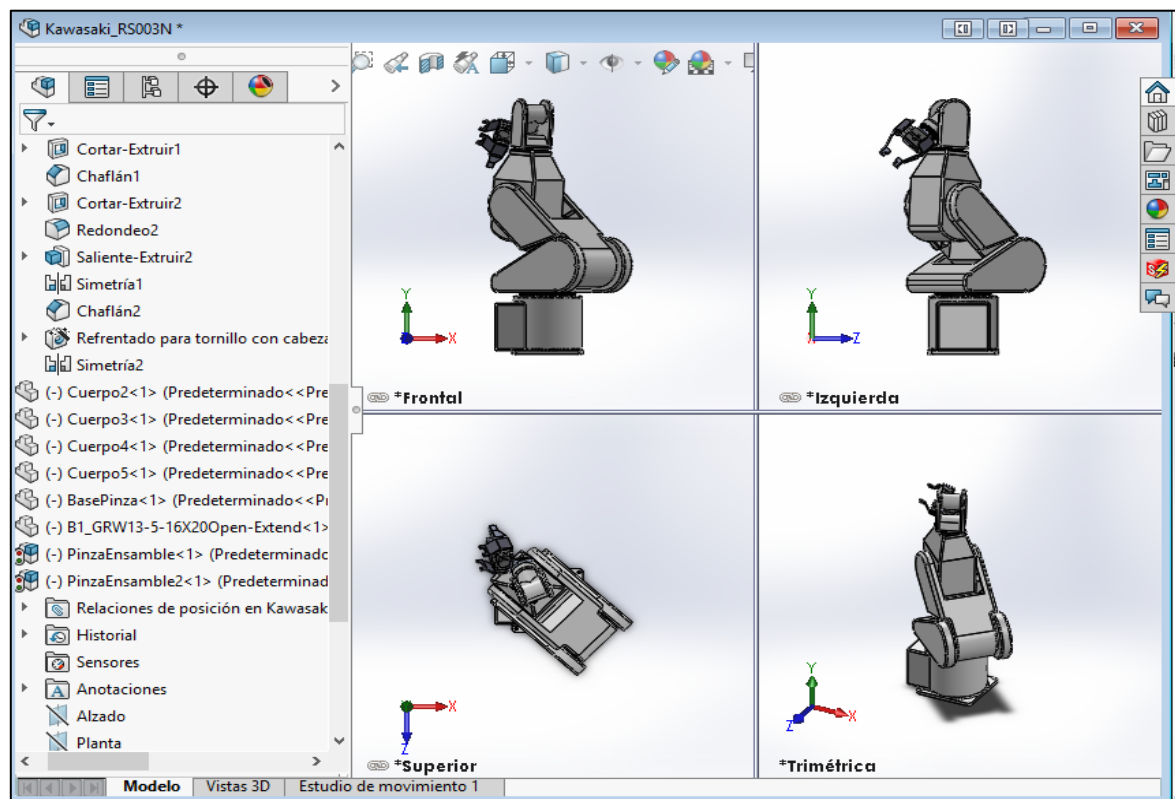


Figura 15-4: Vista del brazo robótico completo

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

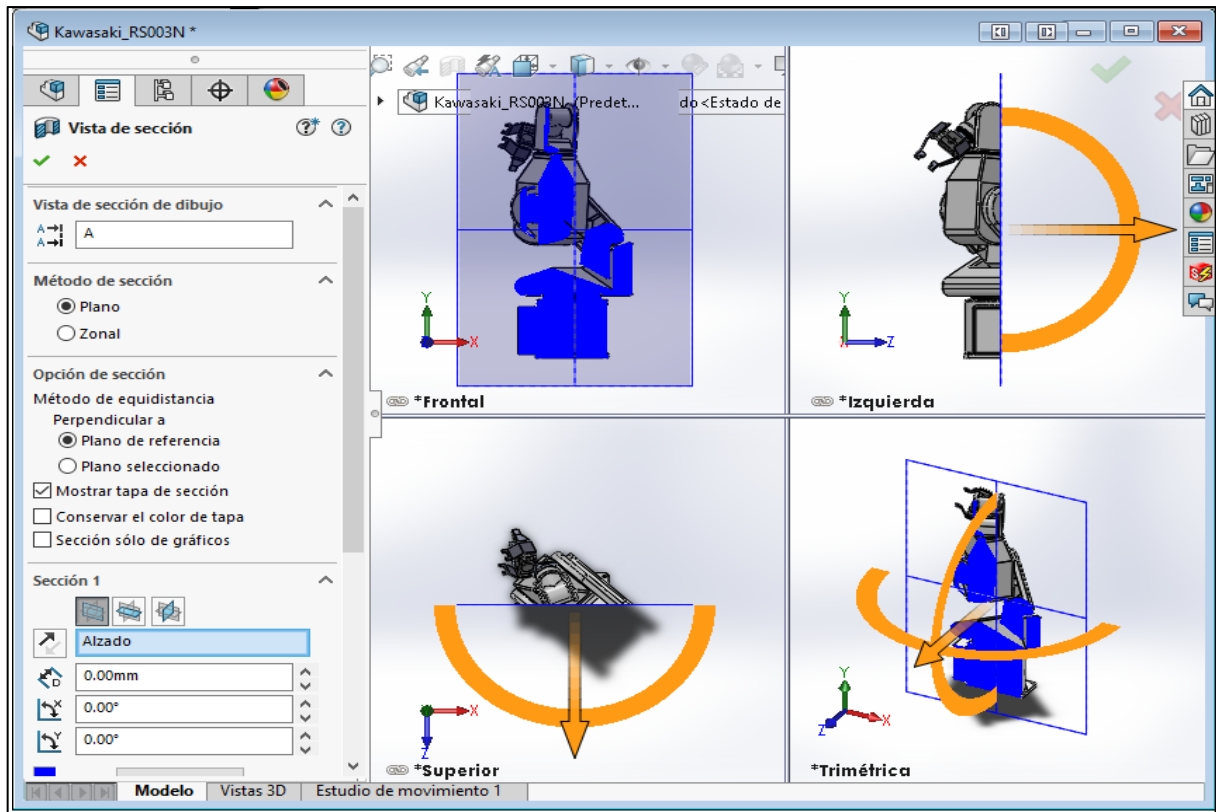


Figura 16-4: Vista en 3D de los diferentes ángulos del brazo robótico

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.6. Diseño de la Estructura del brazo robótico en Simulink por medio de la herramienta SimMechanics.

La mecánica del robot Kawasaki RS 003N, guarda cierta similitud con la anatomía del brazo humano. El movimiento entre las articulaciones es de rotación. Éste movimiento está limitado a los 180° de rotación de los servomotores utilizados. Al poseer seis grados de libertad (GDL), dentro de las configuraciones más frecuentes en robots industriales podemos clasificarlo como robot angular o antropomórfico.

SimMechanics, es una herramienta muy útil que extiende las posibilidades de modelado que ofrece Simulink, permitiendo el modelado CAD del brazo robótico, mediante bloques que representan cuerpos, articulaciones, controladores, actuadores y sensores. Bloques, a partir de los cuales, SimMechanics resuelve las ecuaciones de movimiento del sistema objeto de estudio y provee un

entorno de simulación multicuerpo en 3D. El modelo en 3D de las piezas del robot Kawasaki RS 003N, es importado desde el software CAD Solid-Works.

4.6.1. Bloques Usados de SimMechanics

Dentro de los bloques de SimMechanics empleados en el proyecto se encuentran:

- ❖ **World, Mechanism Configuration y Solver Configuration.** Estos tres bloques fueron fundamentales porque permitieron el ensamblaje de los componentes del sistema objeto de simulación; el World, sirve como referencia general del sistema; el Mechanism Configuration se encarga de configurar la fuerza de la gravedad; y, al Solver Configuration se lo emplea para especificar los parámetros de cálculo de la simulación.
- ❖ **Solid.** Por cada una de las piezas del robot Kawasaki RS 003N exportada desde SolidWorks existe un bloque Solid. En este bloque se insertan los detalles y características geométricas de cada pieza como un archivo con extensión .STL como se muestra en la figura 4. Así también en este bloque se detallan parámetros de masa, centro de masas, inercia, etc.
- ❖ **Reference Frame.** Este bloque ubica un sistema de referencia en el punto de enlace.
- ❖ **Transform.** Fue empleado para establecer las relaciones de posición, traslación y rotación entre los eslabones conectados a los dos extremos del bloque, disponiendo de forma adecuada los componentes físicos del sistema.
- ❖ **Join.** Son bloques de unión que se encargan de establecer la relación entre los puntos conectados a sus puertos F y B. Un join puede definir señales de fuerza/torque, posición lineal/angular, velocidad y aceleración.
- ❖ **RootGround:** Define la Referencia del Sistema de Coordenadas, dando valores a las variables (x), (y) y (z); de acuerdo a la posición deseada por el usuario
- ❖ **Revolute:** Representa una Conexión de un grado de libertad Rotacional.
- ❖ **Prismatic:** Representa una Conexión de un grado de libertad de Traslación.

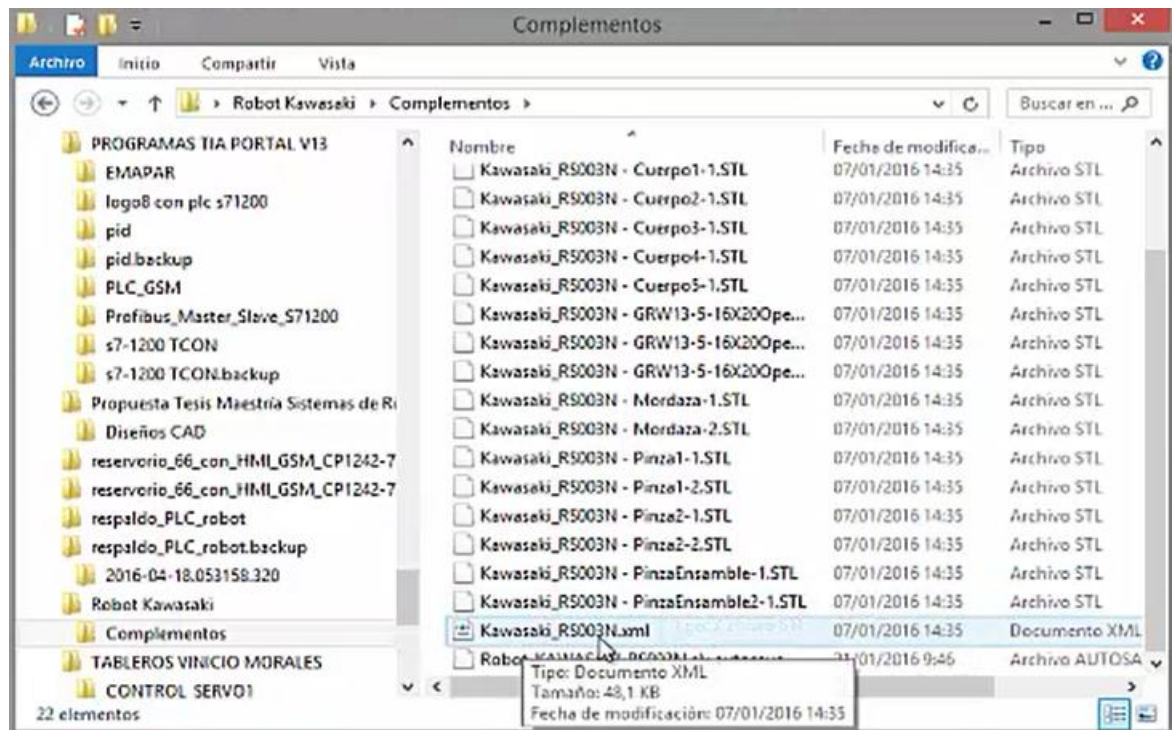


Figura 17-4: Bloques Solid de las piezas del Kawasaki RS 003N

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

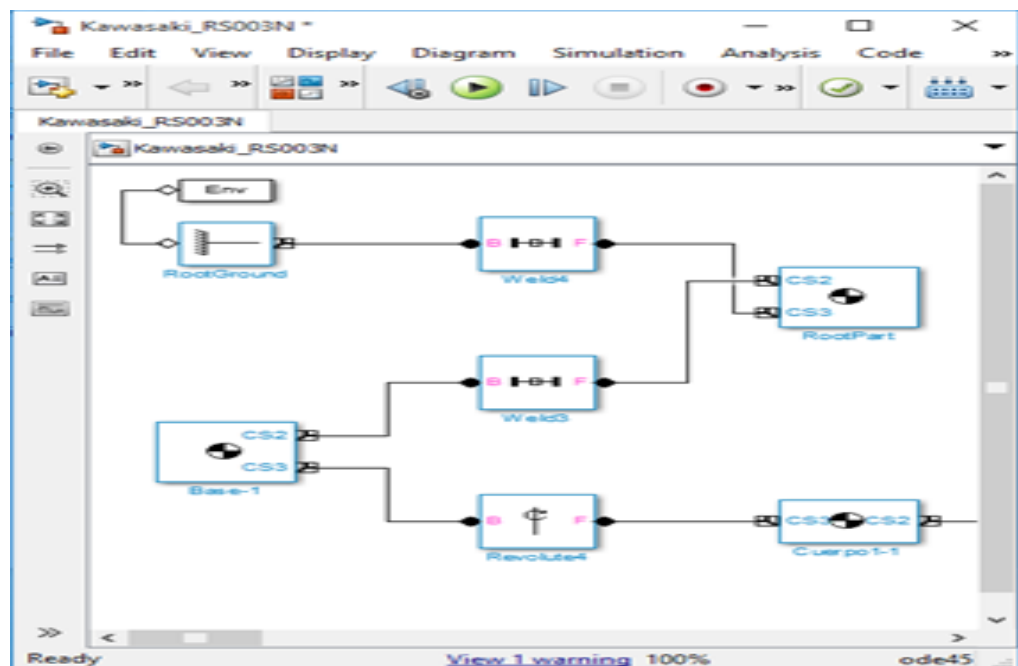


Figura 18-4: Bloques de la Librería SimMechanics

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.7. Exportación de SolidWorks a Simulink

El proceso de exportación de los objetos 3D de SolidWorks 2016 a SimMechanics, se realizó en primera instancia la instalación SimMechanics Link, una herramienta The MathWorks Inc, compatible con MATLAB 2016 a y con SimMechanics.

Una vez instalado el software, se procedió utilizando el menú desplegable de SimMechanics y seleccionando en la opción Export, SimMechanics First Generation. Posteriormente se guardaron los archivos que se crearon al exportar en la misma carpeta en la que se integran los componentes del ensamblaje de SolidWorks, esto limita la presencia de fallos en el sistema.

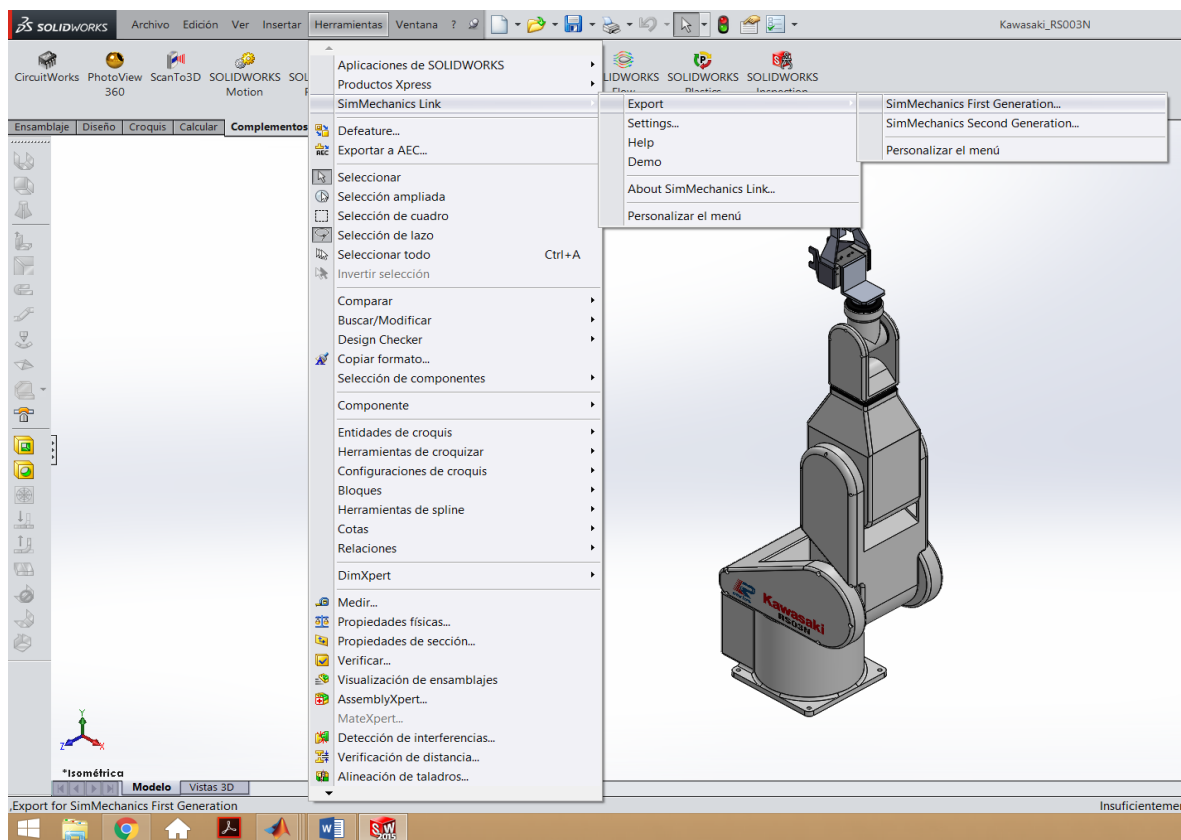


Figura 19-4: Exportación de Solidworks a Simulink

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Ya generados los archivos del ensamblaje se procedió a importarlos a MATLAB, para lo cual se seleccionó la carpeta donde se encontraban guardados los ficheros como carpeta de trabajo, luego se

ejecutó el comando `mech_import` ['Kawasaki_RS003N.xml']. Inmediatamente se abrió una ventana de Simulink y con ello dio inicio el proceso de importación.

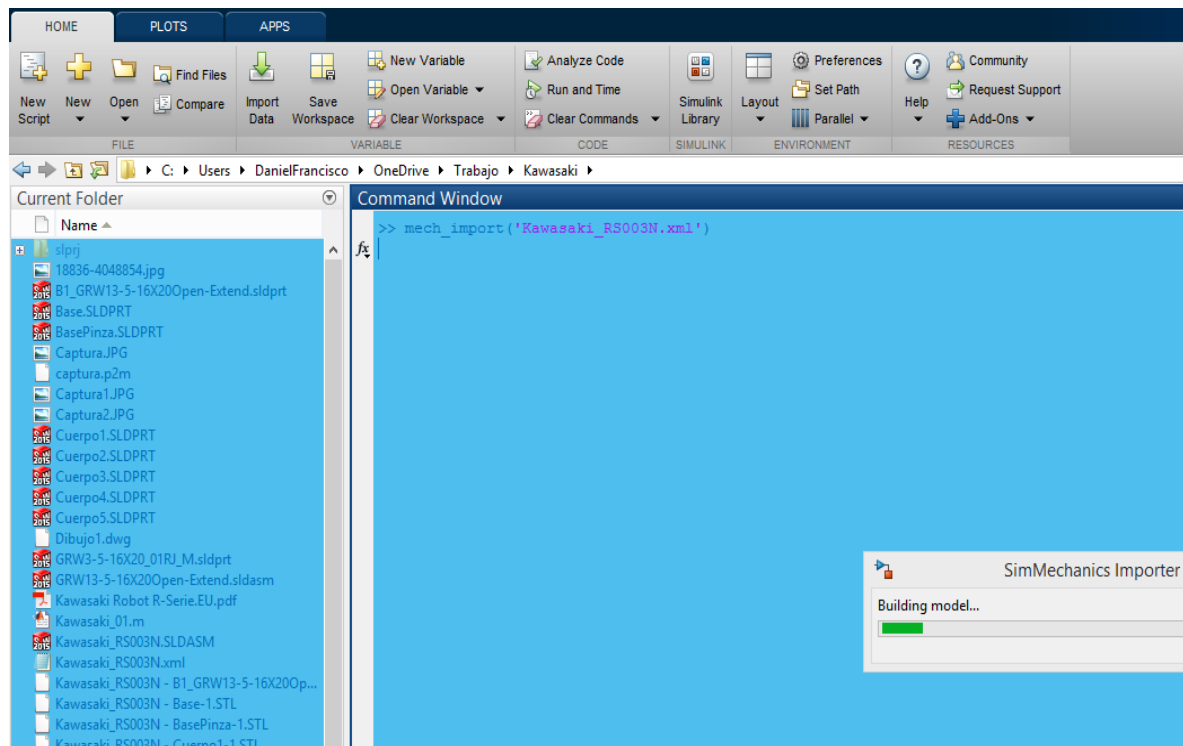


Figura 20-4: Importación a Matlab

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

El procedimiento finalizó con la creación de un modelo en 3D del robot en Simulink, de las piezas del robot Kawasaki RS 003N, con bloques de SimMechanics, análogo al modelo exportado. Una vez realizado estos procedimientos, el modelo del brazo robótico se genera en Simulink de manera automática.

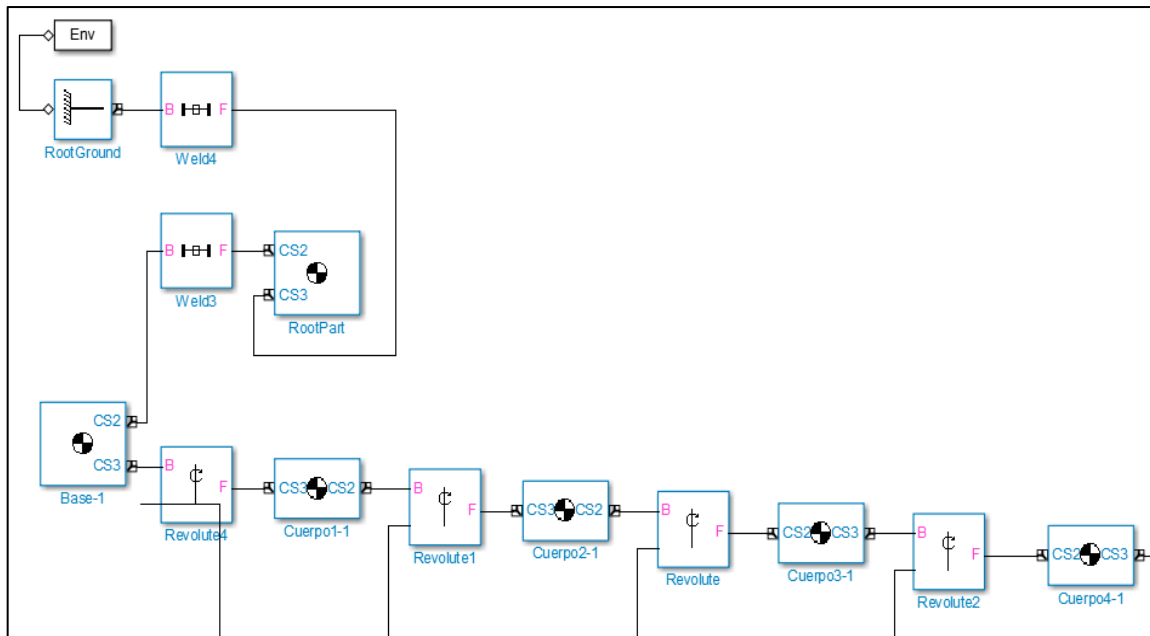


Figura 21-4: Generación del modelado en Simulink

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Generado el modelo en Simulink, se procedió a realizar el modelo del control de las *Revolute* y *Prismatic* con *Join Actuator*.

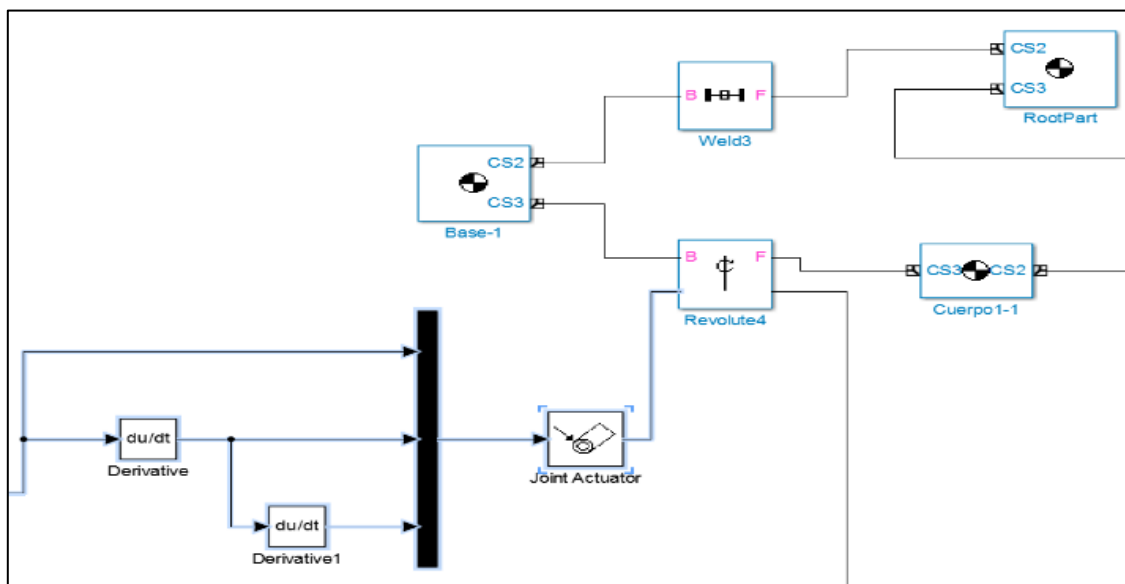


Figura 22-4: Modelo del control de las *Revolute* y *Prismatic* con *Join Actuator*

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

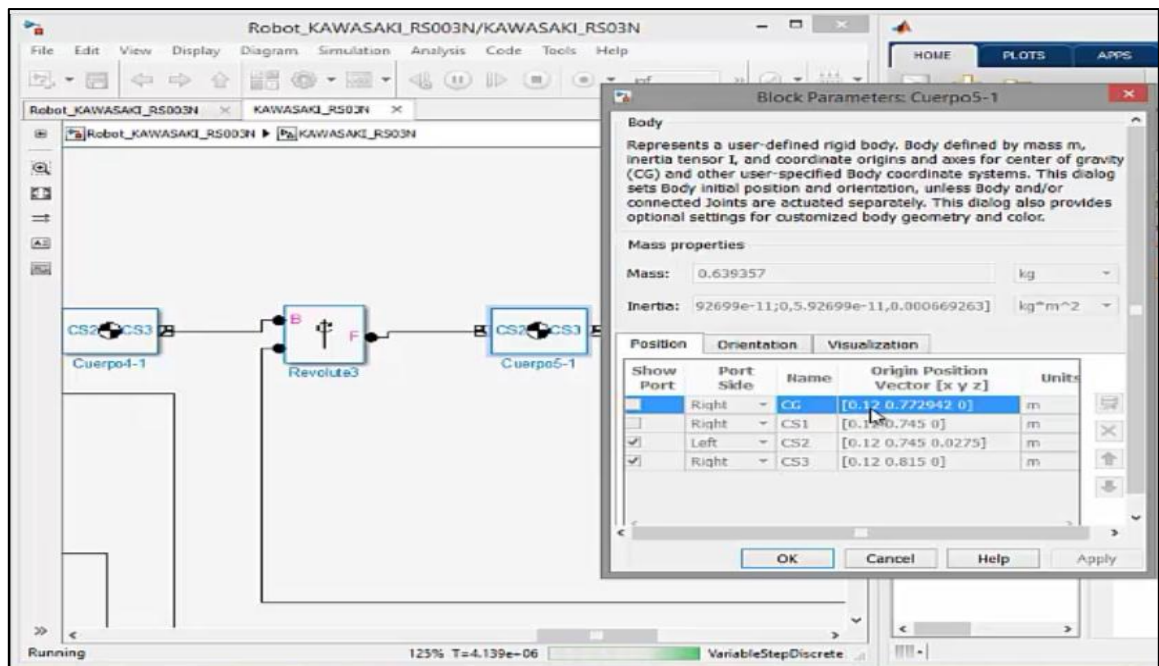


Figura 23-4: Parámetros de los bodys

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Se genera un archivo considerando los topes en los vectores de cada uno de los eslabones y articulaciones, asignando los valores de las salidas de puerto y las posiciones originales de los vectores (x) y (y). Esto crea la posibilidad de mediante un clic en la pantalla, mover el robot y mostrar una posición adecuada que permita visualizar el movimiento de cada una de las piezas del Kawasaki RS 003N.

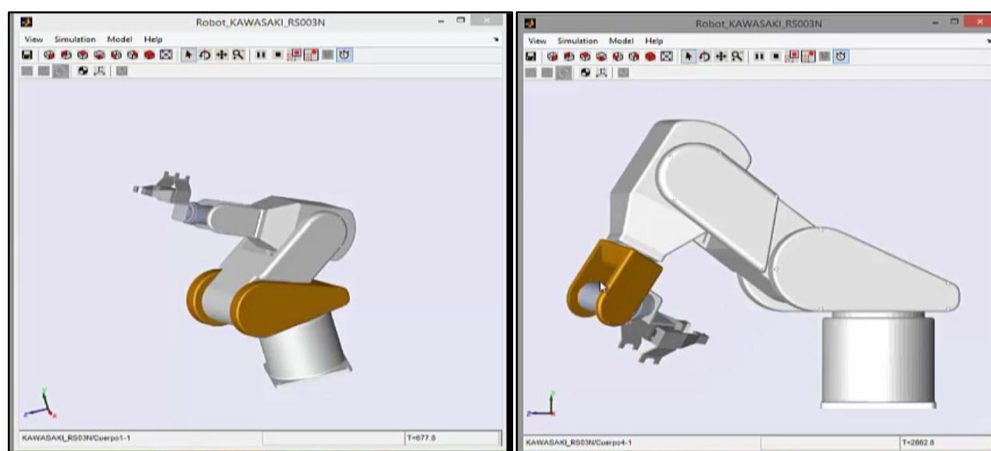


Figura 24-4: Aspecto de la animación 3D del simulador del robot Kawasaki

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.8. Modelado del Sistema de Control

4.8.1. Control del movimiento del robot sin el uso de algoritmo

Se realizó el modelamiento del sistema de control de los ángulos del Robot Kawasaki RS 003N con las barras que mueven cada uno de los ejes y la pinza. Los valores asignados a los ángulos tope de las juntas se establecieron en base a la hoja o catálogo de datos del robot, considerando vectores (x) y (y), estableciendo posiciones máximas y mínimas de cada una de las juntas en Matlab.

Tabla 2-4: Ángulos máximos y mínimos establecidos para las juntas

Articulación	Ángulo mínimo	Ángulo por defecto (Matlab)	Ángulo máximo
JT1	-160°	0°	160°
JT 2	-60°	0°	150°
JT 3	-120°	0°	150°
JT 4	-360°	0°	360°
JT 5	-135°	0°	135°
JT 6	-360°	0°	360°
Pinza (robot real)	-16°	-16°	0°

Fuente: Investigación de campo

Elaborado por: Irma Villa 2017

Una vez definidas las posiciones máximas y mínimas de cada junta, es preciso observar en la pantalla de Matlab el robot y las diferentes juntas, de forma manual se pueden deslizar la barra de posición de cada junta.

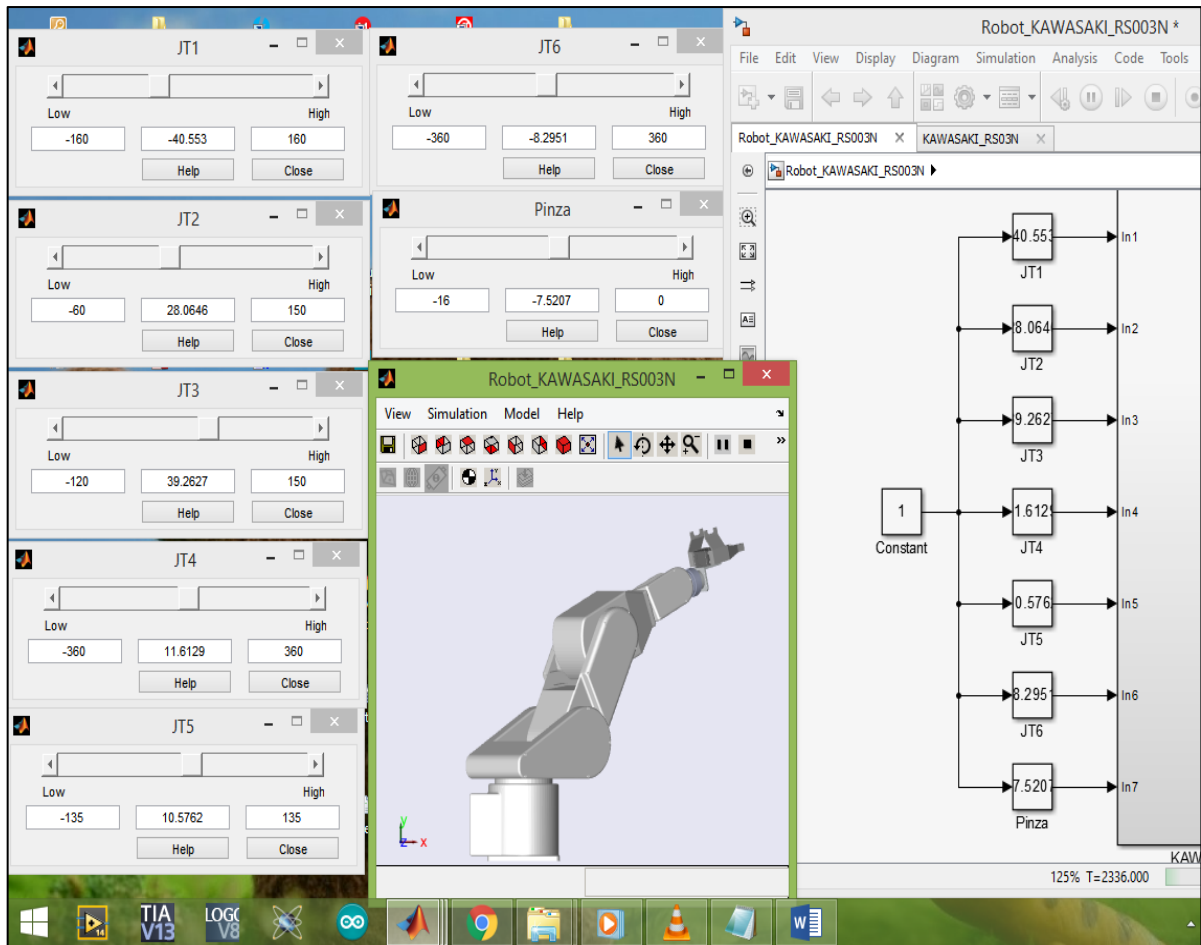


Figura 25-4: Modelado del sistema de control de los ángulos del robot

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Con los movimientos generados se verifica el movimiento del robot mediante un control manual realizado que consiste en dar click en las flechas para mover los ejes del robot de acuerdo con la necesidad de posicionamiento de cada una de las piezas. Con este procedimiento, se visualiza en tiempo real, el movimiento de cada una de las piezas del robot, se muestra además la posición de origen del robot, el radio de trabajo y la posición de la pinza.

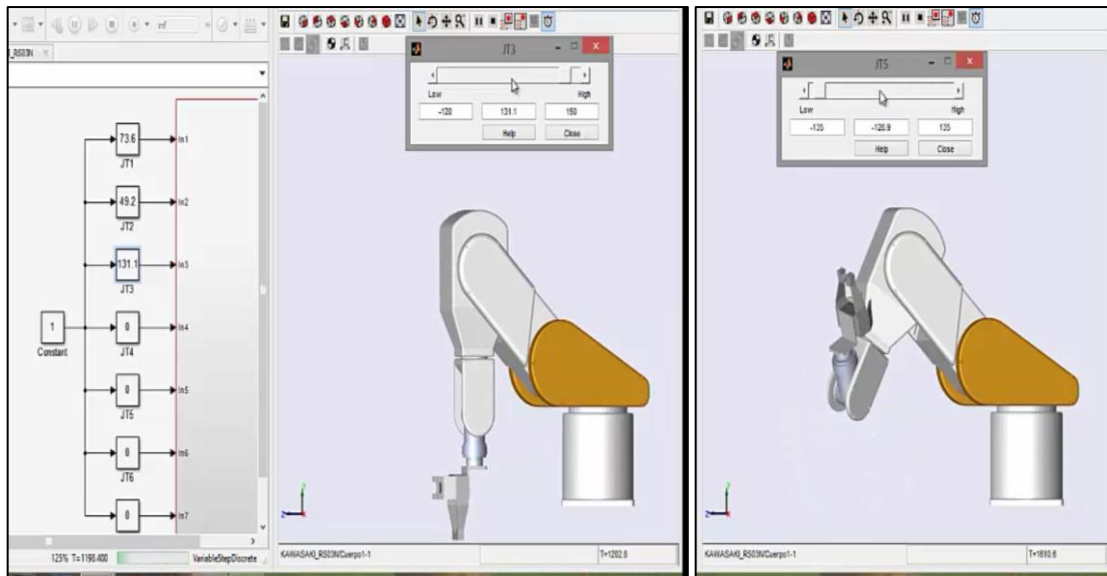


Figura 26-4: Despliegue del movimiento de las juntas 3 y 5

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

La pinza tiene un movimiento prismático que permite la apertura y cierre de la misma. Cabe recalcar que el modelo CAD se desplaza en función de los movimientos mecánicos reales del robot.

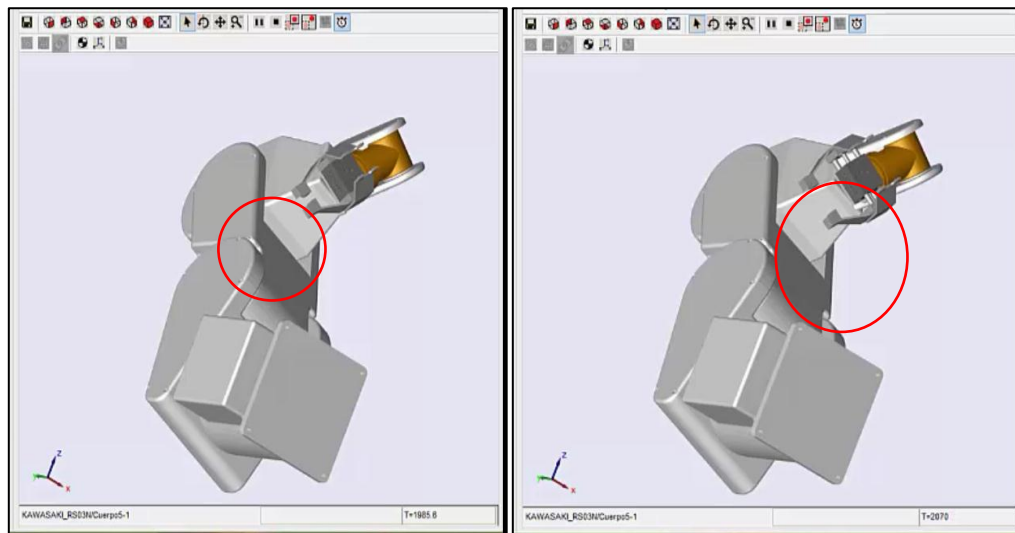


Figura 27-4: Despliegue del movimiento de pinza (juntura 7)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Una vez establecidos los valores de los vectores (x) y (y), al iniciar Matlab, se procedieron a cargar las sliders con los valores asignados a las articulaciones.

4.8.2. Control del movimiento del robot mediante el algoritmo Denavit - Hartenberg

Recordemos que para la simulación se ha empleado el Software Simulink, un simulador en 3D que posibilita la visualización de la posición y orientación del brazo robótico a través de la pantalla del ordenador y que incluye tres aspectos:

- ❖ Una representación gráfica en tres dimensiones del posicionamiento y orientación del Kawasaki RS 003N.
- ❖ Una visualización en pantalla de los datos correspondientes a los ángulos de los ejes del robot, donde se definen posiciones extremas o topes.
- ❖ La posibilidad de dar movimiento a las piezas del robot de forma manual.

A partir de ahora es necesario realizar un análisis del movimiento del robot a fin de obtener una descripción analítica del movimiento espacial del robot para llevar a cabo el guiado y el control de los movimientos del robot y de los resultados de simulación obtenidos. Con ello se pretende conseguir que el usuario establezca las coordenadas (x), (y) y (z) de referencia y el robot en 3D adopte la posición y orientación deseada.

4.8.3. Parametros de Denavit - Hartenberg

Para el control automático de los movimientos del Kawasaki RS 003N se ha utilizado el algoritmo Denavit - Hartenberg, un algoritmo que proporciona el control al usuario sobre los movimientos de los diferentes ejes del robot. En el sistema establecido el usuario puede ejecutar acciones de control, el programa las recibe, representa en pantalla la nueva posición y orientación del robot dando la muestra al usuario. Lo cual permite que el usuario pueda operar el robot 3D visualizándolo en el simulador.

4.8.3.1. Ubicación de la variables articulares

En primera instancia se ha definido la ubicación de la variables articulares, que para el robot Kawasaki RS 003N son las que se muestran en la figura 29-4. Para la obtencion de los giros se ha aplicado la regla de mano derecha (en la práctica esto se puede modificar en la interfaz de usuario para que coincida con el modelo).

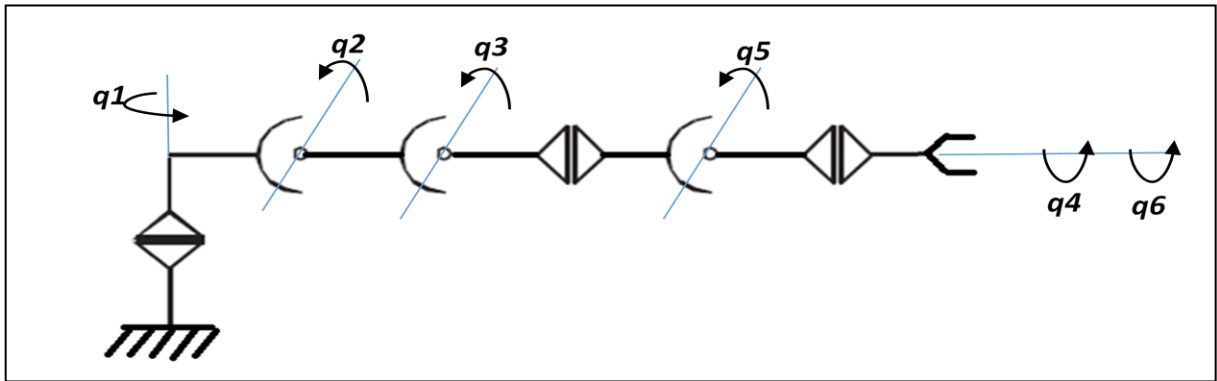


Figura 28-4: Variables articulares del robot Kawasaki RS 003N

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.8.3.2. Modelo cinemático directo usando la librería de Peter Corke.

La librería Peter Corke, permite modelar mediante hilos un manipulador robótico, así como calcular su cinemática directa mediante los parámetros de Denavit – Hartenberg y la inversa mediante algoritmos recursivos. El detalle de la librería, así como la totalidad de sus funciones puede encontrarse en <http://petercorke.com/wordpress/toolboxes/robotics-toolbox/>. La vista inicial es decir la posición de offset se encuentra en la figura 30-4.

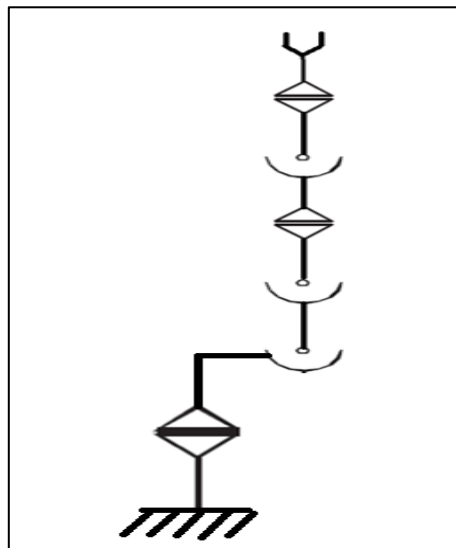


Figura 29-4: Modelo cinemático directo usando la librería de Peter Corke.

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.8.3.3. Control del movimiento del robot a través de cinemática directa

Se realizó el ingreso de los valores máximos y mínimos de las juntas (q_1 , q_2 , q_3 , q_4 , q_5 , p_z), a fin de establecer los rangos de movimiento del robot.

```
set(handles.sq1,'Value',0)
set(handles.sq1,'max',160)
set(handles.sq1,'min',-160)

set(handles.sq2,'Value',0)
set(handles.sq2,'max',150)
set(handles.sq2,'min',-60)
```

Posteriormente, se procede a declarar en Matlab las variables de las articulaciones como globales, se visualizan los valores de los sliders y se los asigna a cada variable, transformando así a radianes los valores establecidos en grados. Con estos datos se calculan los valores de los ejes (x), (y) y (z).

```
global q1 q2 q3 q4 q5 q6 pz
assignin('base','q1',q1)
assignin('base','q2',q2)
assignin('base','q3',q3)
assignin('base','q4',q4)
assignin('base','q5',q5)
assignin('base','q6',q6)
assignin('base','pz',pz)
```

Una vez identificado el modelo del robot en Simulink, mismo que posee una extensión .slx, con la ayuda del comando **set_param** (), se modifican los ángulos en los sliders de acuerdo a los datos ingresados para la simulación.

```

find_system('Robot_KAWASAKI_RS003N');
set_param('Robot_KAWASAKI_RS003N/Constant', 'Value', num2str(
set_param('Robot_KAWASAKI_RS003N/Constant1', 'Value', num2str
set_param('Robot_KAWASAKI_RS003N/Constant2', 'Value', num2str
set_param('Robot_KAWASAKI_RS003N/Constant3', 'Value', num2str
set_param('Robot_KAWASAKI_RS003N/Constant4', 'Value', num2str
set_param('Robot_KAWASAKI_RS003N/Constant5', 'Value', num2str
set_param('Robot_KAWASAKI_RS003N/Constant6', 'Value', num2str

```

Para la creación de robot para la visualización en diagrama de líneas se utilizan los comandos de la Librería Peter Corke y la función “fnccrearrobot”, considerando los parámetros de Denavit - Hartenberg. Mediante el comando de SerialLink pueden acoplarse todos los eslabones, en este caso los eslabones se consideran solamente con dimensión, pero también existe la posibilidad de incluir masa, centros de gravedad, y demás parámetros dinámicos.

```

function fnccrearrobot(q1,q2,q3,q4,q5,q6)
cla reset
L(1)=Link([0 0 0 0 0]);
L(2) = Link([0 0.25 0.12 pi/2 0]);
L(3) = Link('revolute','offset',pi/2, 'd', 0, 'a', 0.25, 'al
L(4) = Link('revolute','offset',-pi/2, 'd', 0, 'a', 0, 'alph
L(5) = Link([0 0.25 0 pi/2 0]);
L(6) = Link([0 0 0 -pi/2 0]);
L(7) = Link([0 0.08 0 0 0]);

global limpiador
pos=[0, q1, q2, q3,q4,q5,q6];
limpiador=SerialLink(L,'name','Kawasaki RS03N');
limpiador.plot(pos)

```

La función mover el robot “fncmoverrobot”, lo grafica en una nueva posición de acuerdo con los valores que se mueven en los sliders.

```

global q1
q1=deg2rad(get(handles.sq1,'Value'));
set(handles.vsq1,'String',rad2deg(q1));
fncmoverrobot()
find_system('Robot_KAWASAKI_RS003N');
set_param('Robot_KAWASAKI_RS003N/Constant'

```


4.8.3.4. Control del movimiento del robot a través de cinemática inversa

Para la cinemática inversa se realiza el ingreso de los valores de los ejes (x), (y) y (z), para obtener los valores de las juntas (q_1 , q_2 , q_3 , q_4 , q_5 , p_z).

```
global   limpiador q2 q3 q1 q5 q4 q6
q4=deg2rad(get(handles.sq4,'Value'));
set(handles.vsq4,'String',rad2deg(q4));
set(handles.etvsq4,'String',rad2deg(q4));
fcnmoveerrobot()

pos=[q1 q2 q3 q4 q5 q6]
CD=fkine(limpiador,pos)
set(handles.etcx,'string',CD(1,4));
set(handles.etcy,'string',CD(2,4));
set(handles.etcz,'string',CD(3,4));
```

La creación de robot para la visualización en diagrama de líneas considera los comandos de la Librería Peter Corke y la función “fcncrearrobot”, igual que como se procedió en la cinemática directa, igualmente la función “fcnmoveerrobot” permitió dar movimiento al brazo robótico.

Para verificar la correcta definición de los valores de los vectores que se han asignado, se convierten los radianes devueltos por la librería del Peter Corke a grados que están en el rango de entre 0° y 360° , controlando que los ángulos estén dentro del rango permitido para cada articulación, a fin de omitir movimientos que puedan realizarse de manera simulada pero que no pueden ser ejecutados por el robot físico.

```
while ( igrados(1,1)>=360 || igrados(1,1)<=-360 )
    if( igrados(1,1)>0)
        igrados(1,1)= igrados(1,1)-360;
    end
    if( igrados(1,1)<0)
        igrados(1,1)= igrados(1,1)+360;
    end
end

while ( igrados(1,2)>=360 || igrados(1,2)<=-360 )
    if( igrados(1,2)>0)
        igrados(1,2)= igrados(1,2)-360;
    end
    if( igrados(1,2)<0)
        igrados(1,2)= igrados(1,2)+360;
    end
end
```

En la figura 31-4 se muestra la cinemática establecida para la visualización del robot en diagrama de líneas, donde se identifican desde la izquierda hacia la derecha, en primer lugar las barra manipuladoras del posicionamiento del robot, luego el tipo de cinemática utilizada, lo resultados de posicionamiento del robot dados en metros y los ángulos generados por cada una de las juntas.

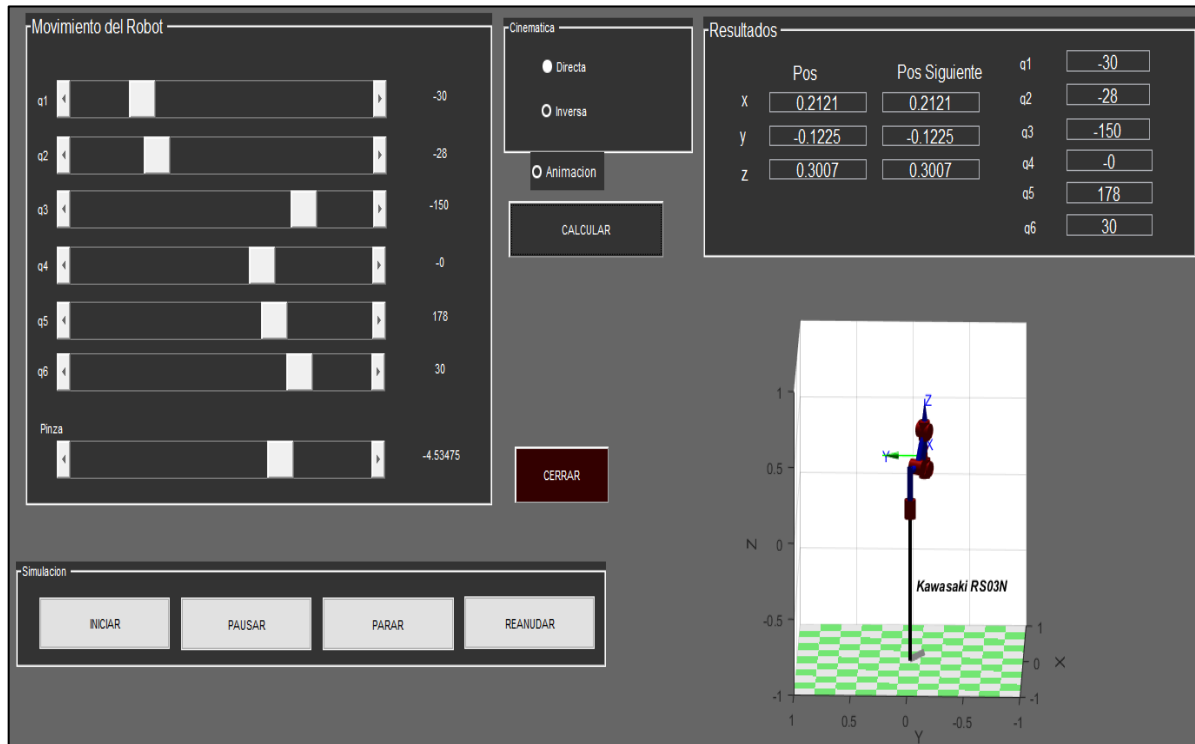


Figura 30-4: Visualización del robot en diagrama de bloques

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

En la figura 31-4 se muestran cinco botones: iniciar, pausar, parar, reanudar y calcular, cuyas aplicaciones son:

- ❖ INICIAR, da inicio a la simulación del Modelo exportado de SolidWorks
- ❖ PAUSAR, pausa la simulación
- ❖ PARAR, produce un stop en la simulación
- ❖ REANUDAR, reinicia la simulación que se ha pausado previamente.
- ❖ CALCULAR, el uso de este botón depende de si se trata de cinemática directa o inversa, en el primer caso, se puede realizar la dinámica directa dados los valores de los ángulos en los edit text, si es inversa se deben ingresar los puntos y calcular los valores de los ángulos en que deben posicionarse las articulaciones.

Finalmente se observa, el diagrama de línea con el posicionamiento de robot ordenado por el usuario, con los vectores posicionales (x), (y) y (z),

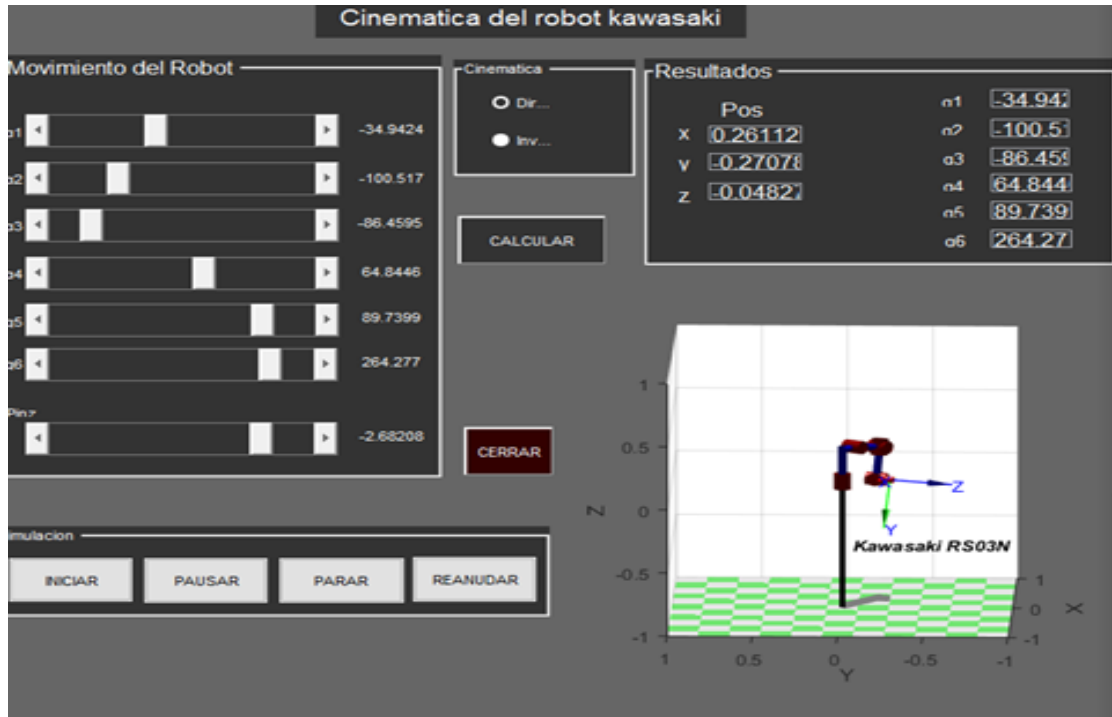


Figura 31-4: Modelo cinemático directo del movimiento del robot

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

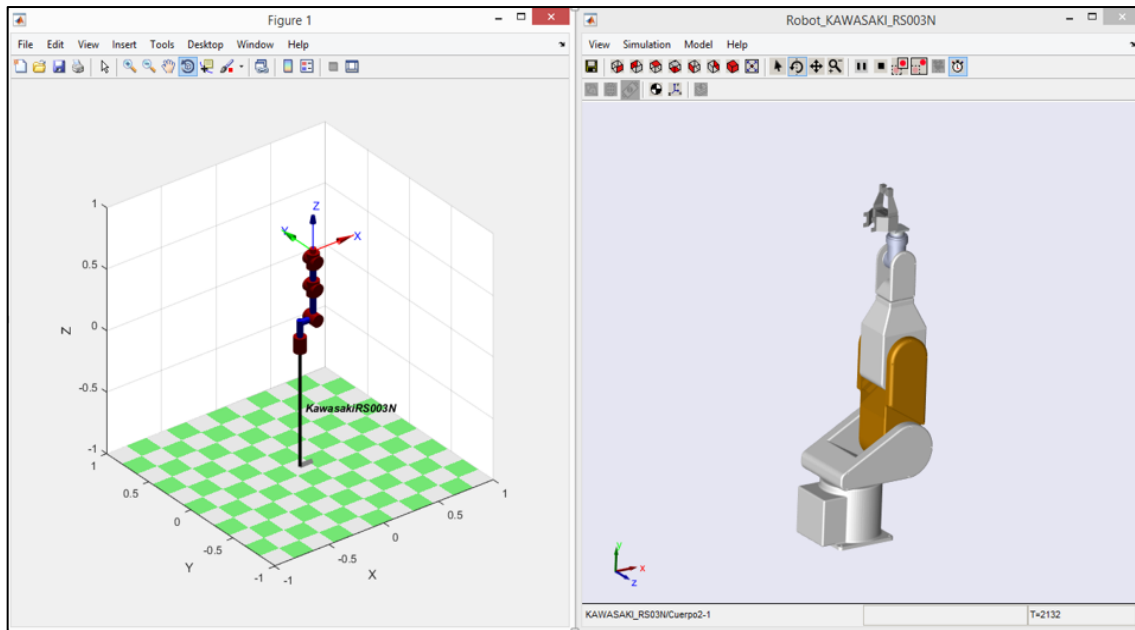


Figura 32-4: Control cinemático directo usando la librería de Peter Corke

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Los ejes no son coincidentes debido principalmente a que en la generación del modelo CAD no se toman en cuenta las leyes del modelamiento cinemático de manipuladores, y es muy común durante la fase de dibujo hacerlo tomando el plano XY como base para dibujo bidimensional y el eje z se considera a la profundidad de los objetos. Para la obtención de modelos matemáticos de manipuladores el eje Z se le considera al que parte perpendicular al suelo, y para la aplicación del método de Devanit-Hartenberg se lo coloca en el eje de rotación de la variable articular en caso de ser rotacional y en caso de ser prismática en el eje donde se produce el desplazamiento (en este robot solo hay rotacionales).

Se puede corregir esto en cierta manera al momento de programar la interfaz manipulando los datos de ingreso para que el modelo obtenido concuerde con el real, o también recurrir a arreglos como una articulación fantasma adicional en la base que rotaría o desplazaría todo el robot.

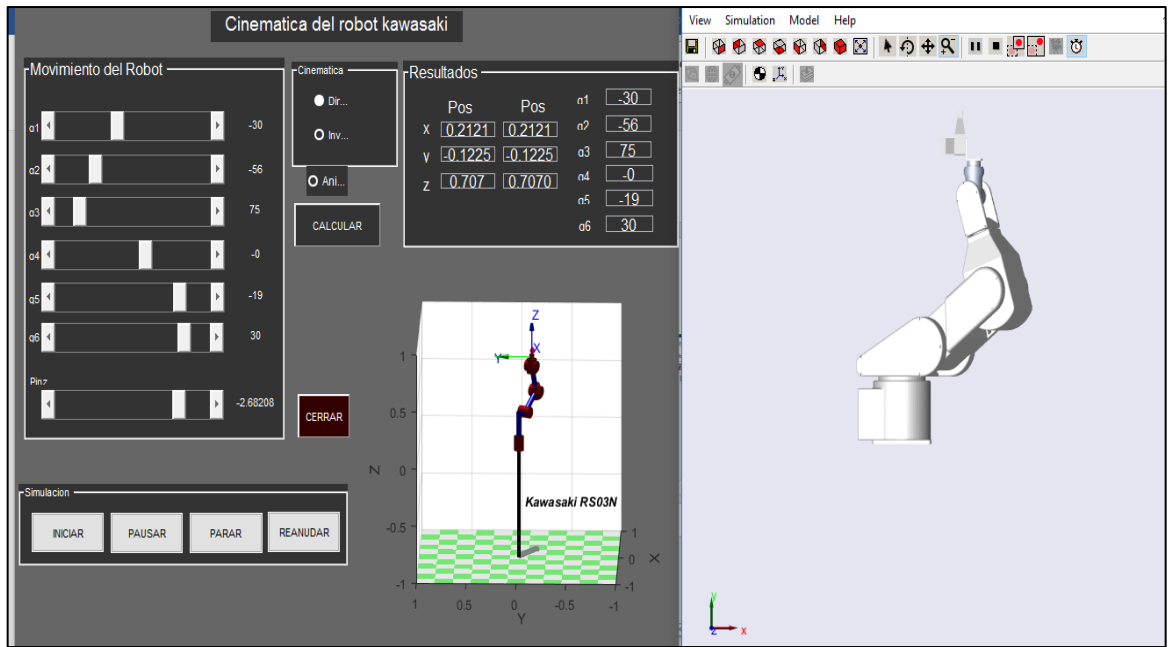


Figura 33-4: Modelo cinemático inverso del movimiento del robot

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

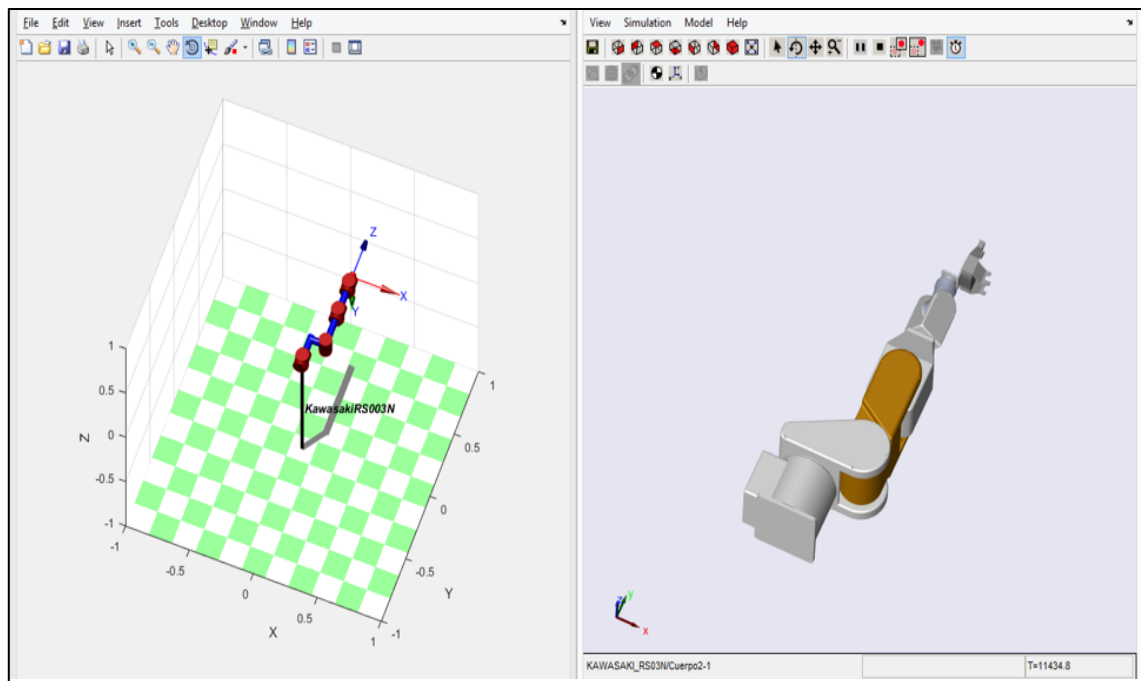


Figura 34-4: Control cinemático inverso usando la librería de Peter Corke

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.9. PRUEBAS DE POSICIONAMIENTO

Con el propósito de comprobar el adecuado funcionamiento del algoritmo de control se realizaron pruebas de posicionamiento del robot físico, en las que el usuario inserta las mismas referencias posicionales tanto en el simulador como en el software del robot físico, para observar si se obtienen en ambos las posiciones finales deseadas. En la práctica se obtiene el mismo resultado en los entornos virtual y físico. Se generaron en la práctica ocho posiciones que se detallan en la tabla.

Tabla 3-4: Cuadro de posiciones ejecutadas con la planta física y la simulación

Juntas	Posición 1	Posición 2	Posición 3	Posición 4	Posición 5	Posición 6	Posición 7	Posición 8
Q1	-0.2	-1.3	-0.2	-22.6	121.4	121.4	3	3
Q2	-0.9	57.9	57.9	57.9	57.9	57.9	57.9	-80.7
Q3	1.5	1.5	-114.6	-114.6	-79.5	33.9	33.9	101.4
Q4	0	0	0	0	0	0	0	0
Q5	-0.099	-0.099	-0.099	-89.19	-89.19	-89.19	-89.19	-10.89
Q6	0	0	0	72	72	72	72	187.2

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

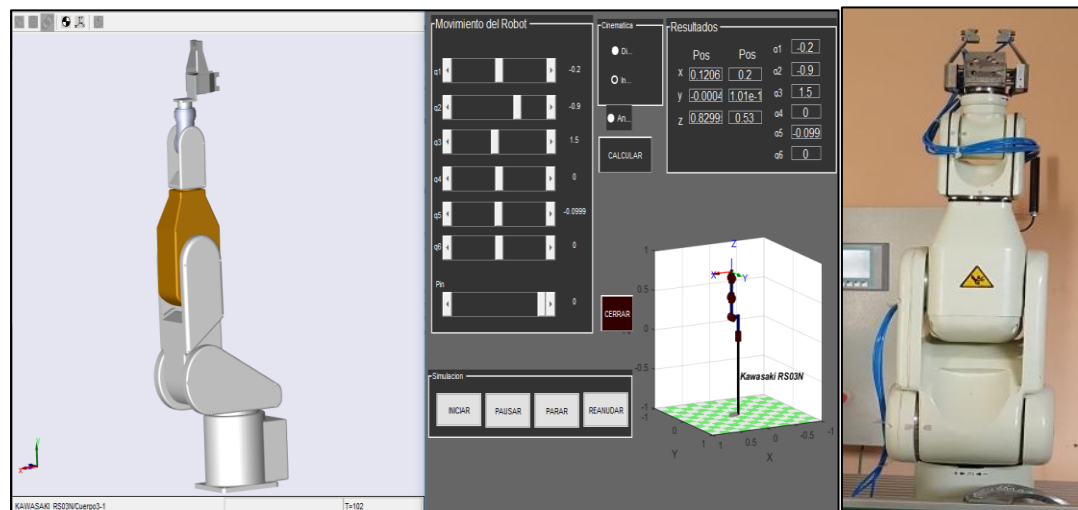


Figura 35-4: Posicionamiento del robot físico y en 3D (posición 0)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

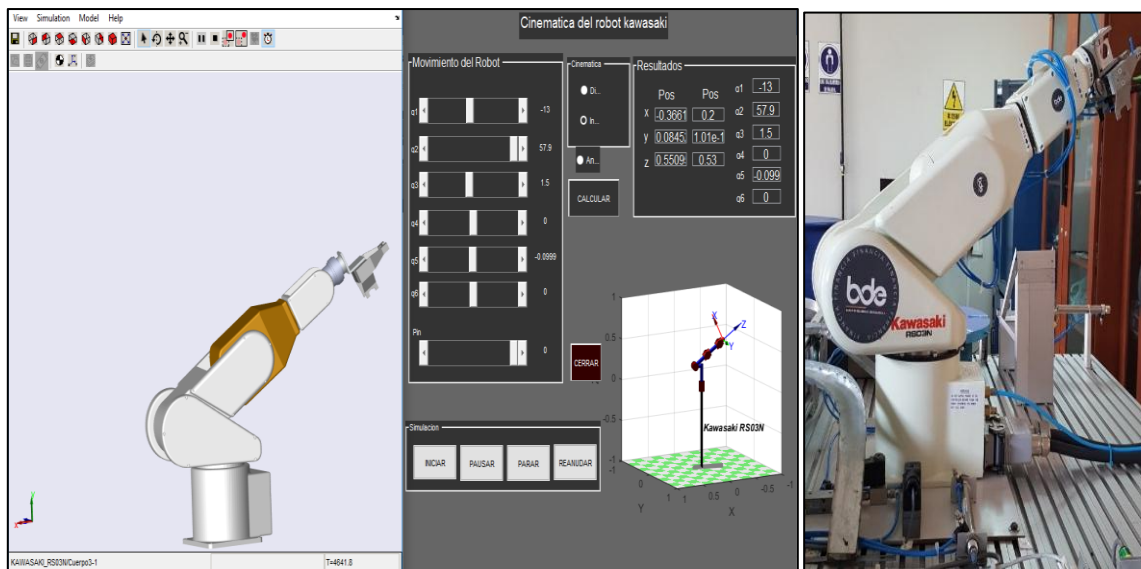


Figura 36-4: Posicionamiento del robot físico y en 3D (posición 1)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

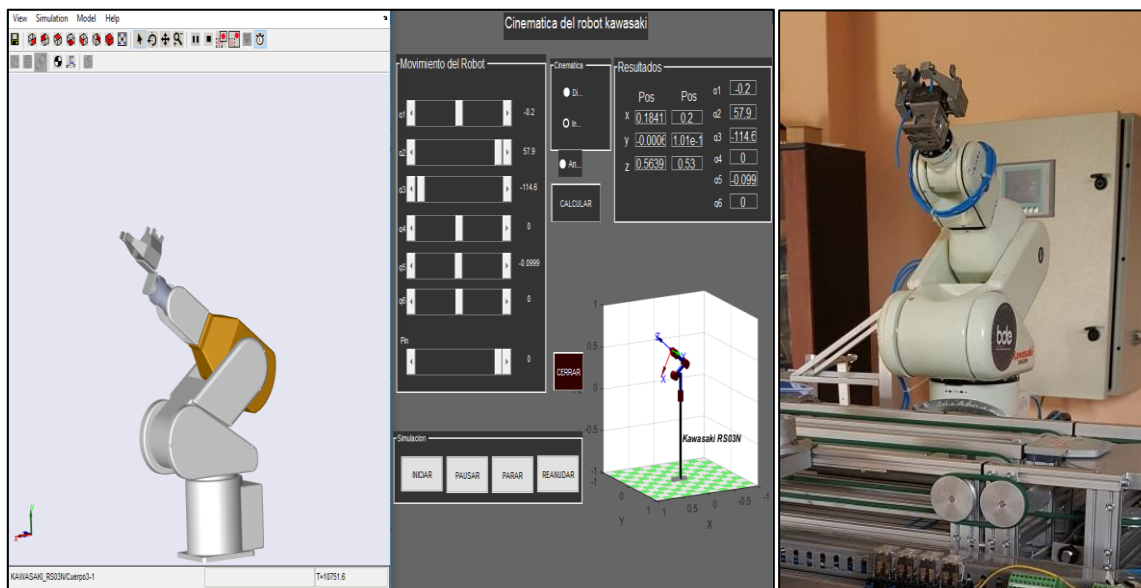


Figura 37-4: Posicionamiento del robot físico y en 3D (posición 2)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

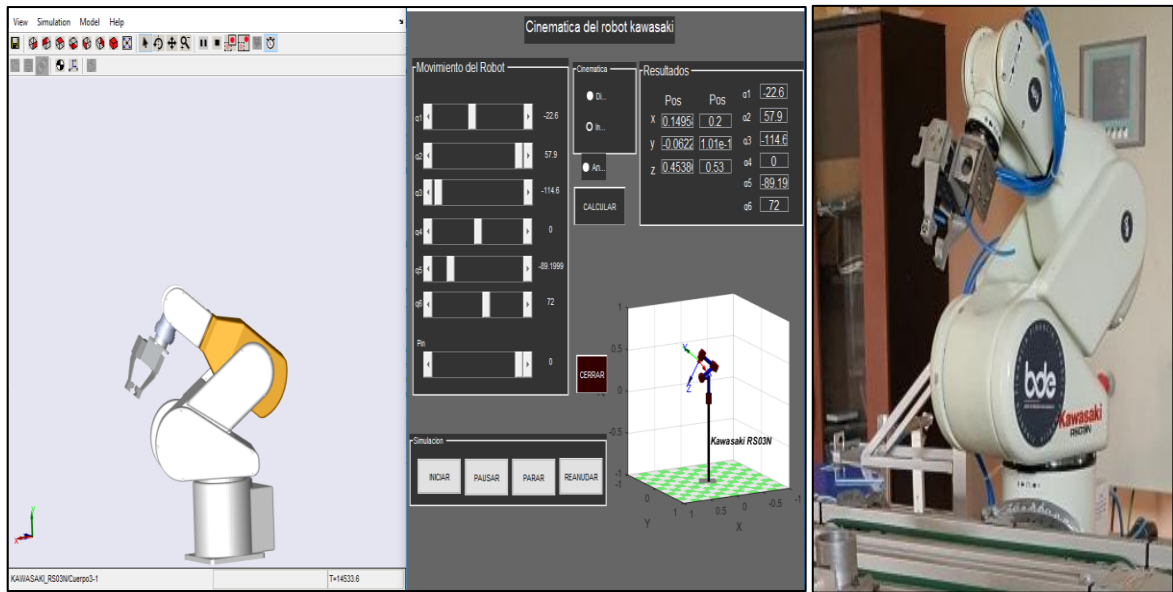


Figura 38-4: Posicionamiento del robot físico y en 3D (posición 3)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

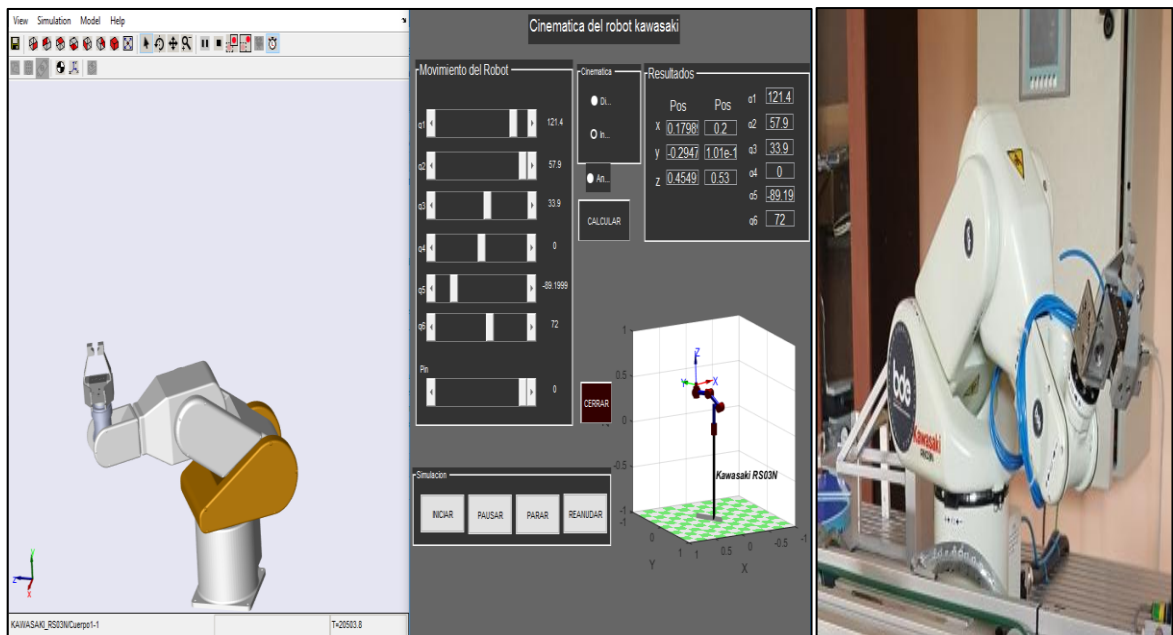


Figura 39-4: Posicionamiento del robot físico y en 3D (posición 4)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

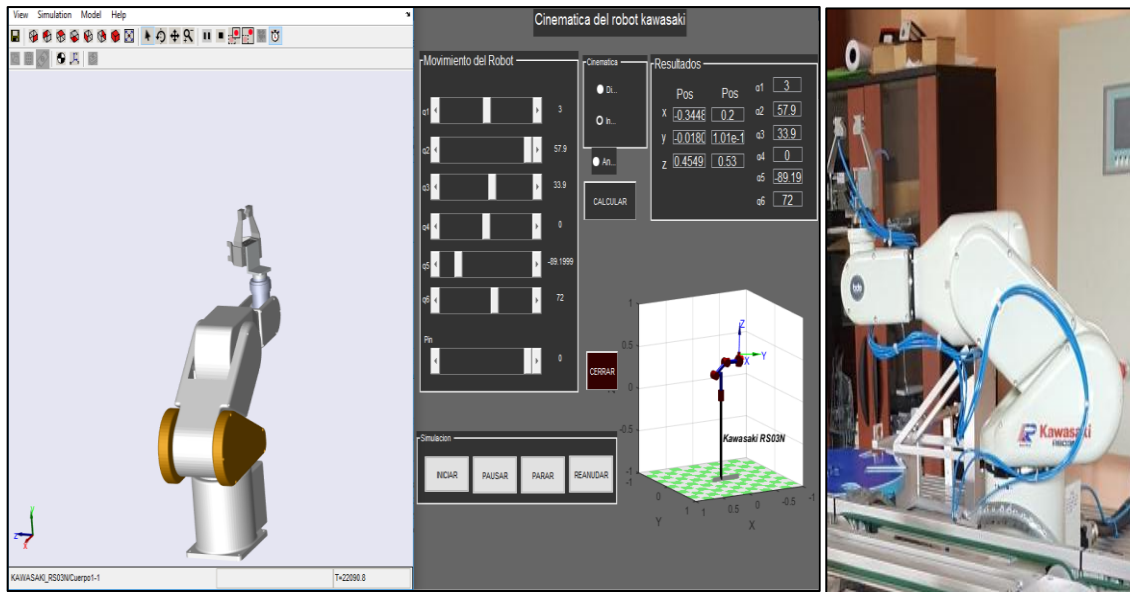


Figura 40-4: Posicionamiento del robot físico y en 3D (posición 5)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

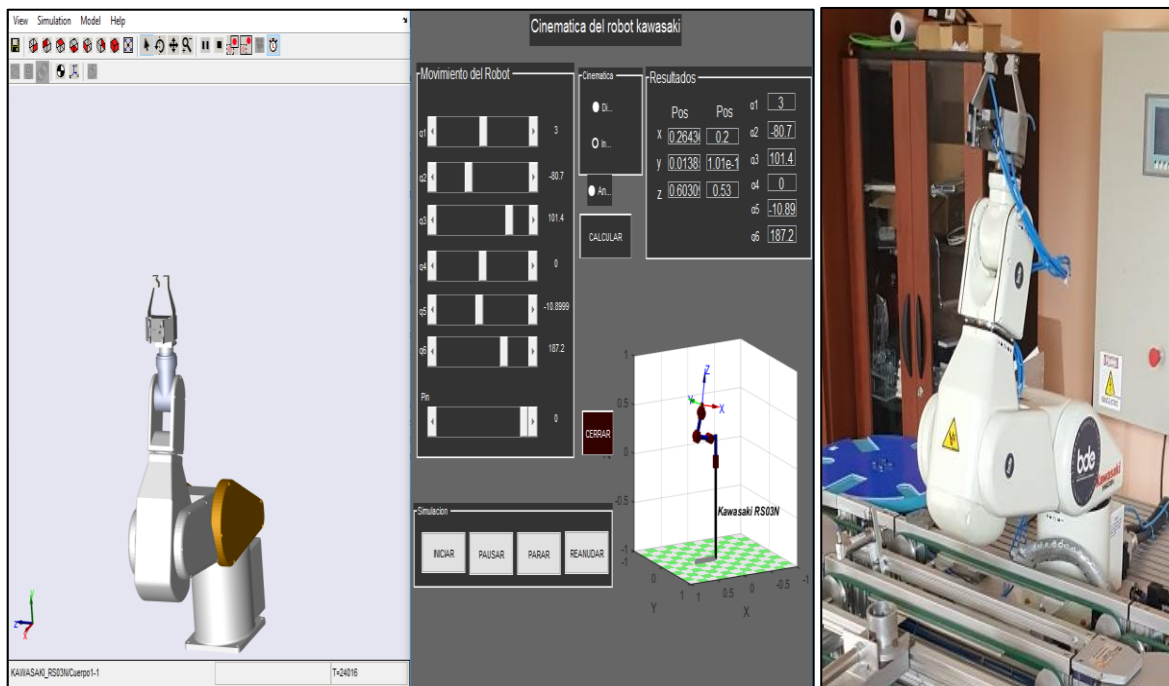


Figura 41-4: Posicionamiento del robot físico y en 3D (posición 6)

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

4.10. RESULTADOS Y DISCUSIÓN

- a. En cuanto al diseño de las piezas, el uso de SolidWorks es muy conveniente por cuanto esta herramienta de modelado CAD, nos permitió diseñar, modificar y crear las piezas en tres dimensiones a demás de manipular de una manera fácil los elementos geométricos disponibles en el software.
- b. El programa de control interpreta adecuadamente las indicaciones del usuario y ordena la nueva posición del robot en el simulador.
- c. La interfaz gráfica empleada para la simulación en 3D, muestra apropiadamente el posicionamiento y la orientación del robot de acuerdo con las órdenes emitidas por el usuario, permitiendo visualizar en pantalla los valores de los ejes y la posición de la pinza o extremo del robot.
- d. Con los resultados obtenidos de las pruebas de posicionamiento, se comprueba que el algoritmo planteado para el control de los movimientos de brazo robótico Kawasaki RS 003N, si funciona.

4.1. Comprobación de Hipótesis

En relación a la hipótesis planteada: *“Es posible realizar el modelamiento, simulación y control del Brazo Robótico KAWASAKI RS 003N, utilizando Solid Works, Simulink de Matlab y el Algoritmo Denavit – Hartenberg”*.

Con SolidWorks es posible realizar el modelamiento de las piezas del Brazo Robótico KAWASAKI RS 003N, con la ayuda de este software se obtiene de una manera fácil y rápida piezas 3D alineadas a las características físicas de cada uno de los eslabones del robot. Con la utilización de figuras geométrica básicas preestablecidas en el SolidWorks se procede a construir parte por parte cada una de las piezas, definiendo para cada una de ellas, las dimensiones exactas según las especificaciones del robot físico.

La construcción del primer eslabón del robot requiere de las figura base cuadrado y círculo, para lo cual se apertura el espacio de trabajo, se selecciona dentro de la pestaña croquis una vista básica y de la barra de herramienta la figura cuadrado circulo que es introducida en el centro del espacio de trabajo. Para definir las medidas del círculo se emplea la opción cola inteligente, posteriormente se procede a dar volumen y color a la figura, dando como resultado el cilindro requerido, como base

de la pieza el resto de la pieza se contruye con figuras básicas como el cuadrado asinando medidas y volumen iguales a las constatadas físicamente. Finalmente cada pieza se guarda en una carpeta predeterminada.

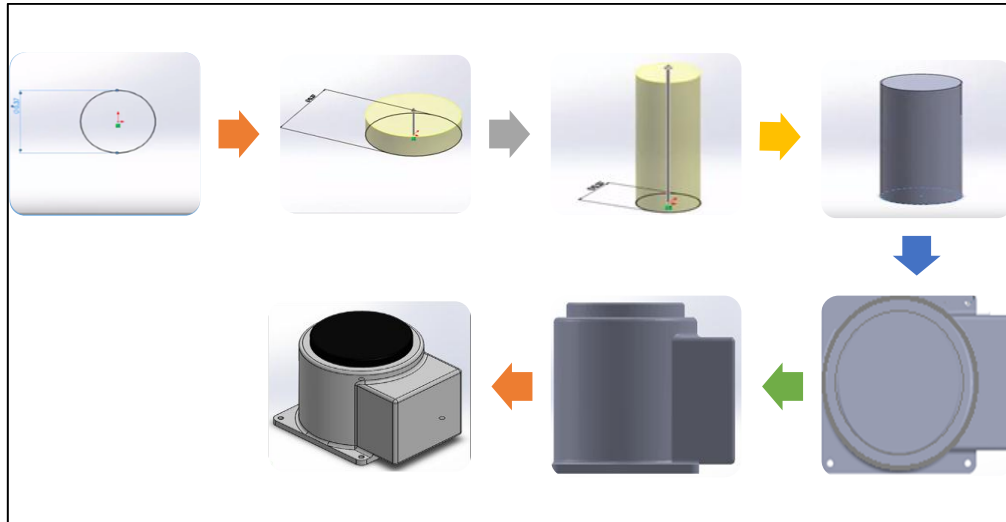


Figura 42-4: Modelado de piezas en Solid Works

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Una vez construidas todas las piezas o eslabones del robot se las ensambla, empleado a opción “ensamble” para generar el modelo el 3D del brazo robótico, como se muestra en la figura.

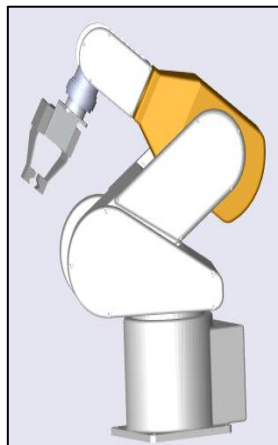


Figura 43-4: Robot Kawasaki RS 003N en 3D

Fuente: Captura de pantalla

Elaborado por: Irma Villa 2017

Con la exportación del modelo CAD al entorno Matlab, se crea el robot 3D en Simulink automáticamente, mediante este entorno se para realizan ejercicios de simulación de los

movimientos del robot, estos movimientos son controlados de forma manual a través del deslizamiento de las posiciones establecidas en cada articulación. Denavit – Hartenberg por su parte se encarga del control automatizado de los movimientos del robot, mediante el uso de un lenguaje de programación, se representa en pantalla las posiciones y orientaciones del robot definidas por el usuario que opera el robot 3D.

Con los resultados obtenidos a trves del desarrollo de la investigación, es posible comprobar la hipótesis plantada al inicio del presente estudio, estableciendo que **SI** es posible realizar el Modelamiento, Simulación y Control del Brazo Robótico KAWASAK 8I RS 003N, utilizando Solid Works, Simulink de Mattlab y el algoritmo Denavit - Hartenberg.

CONCLUSIONES

- ❖ Se ha cumplido el objetivo del Modelamiento del Brazo Robótico KAWASAKI RS 003N, utilizando la herramienta Solidworks, la cual permitió obtener un modelo esperado, considerando las medidas ya optimizadas realizadas por medio de cálculos de Matlab y los parámetros de las figuras base de acuerdo con el diseño físico del robot. Lo primero que se diseñó es la base del robot y posteriormente se procedió con el diseño de los cuerpos uno, dos, tres, cuatro, cinco y la pinza, para lo cual se emplearon figuras bases disponibles en Solidworks, mismas que se acoplaron para dar origen a cada una de las piezas.
- ❖ Para el proceso de exportación de los objetos 3D de SolidWorks a Simulink, fue necesaria la instalación SimMechanics Link, una herramienta The MathWorks Inc, compatible con MATLAB, se procedió utilizando del menú desplegable de SimMechanics la opción Export, SimMechanics First Generation. Posteriormente se guardaron los archivos que se crearon al exportar en la misma carpeta en la que se integran los componentes del ensamblaje de SolidWorks.
- ❖ Una vez generados los archivos del ensamblaje se procedió a importarlos a MATLAB, para lo cual se seleccionó la carpeta donde se encontraban guardados los ficheros como carpeta de trabajo, luego se ejecutó el comando `mech-import` ['Kawasaki_RS003N.xml']. Inmediatamente se abrió una ventana de Simulink y con ello dió inicio el proceso de importación, mismo que finalizó con la creación de un sistema en 3D en Simulink, de las piezas del robot Kawasaki RS 003N, con bloques de SimMechanics, análogo al modelo exportado. Una vez realizado todo este proceso comenzó el trabajo de simulación.
- ❖ El algoritmo Denavit – Hartenberg fue empleado para el control de los movimientos de robot. La librería Peter Corke, permitió modelar mediante hilos un manipulador robótico, así como calcular su cinemática directa e inversa. La creación del robot para la visualización en diagrama de líneas consideró los comandos de la Librería Peter Corke y la función “`fcncrearrobot`”.
- ❖ Con los movimientos generados en el manipulador se verificó el movimiento de los ejes del robot en 3D, de acuerdo con las órdenes de posicionamiento emitidas por el usuario, visualizándose en tiempo real, el movimiento de cada una de las piezas del robot, así como la posición de origen del robot, el radio de trabajo y la posición de la pinza.

RECOMENDACIONES

- ❖ Se recomienda la utilización de Solidworks, pues se trata de una herramienta amigable y efectiva para obtener un adecuado modelamiento de las piezas de robot, para lo cual se debe realizar previamente una adecuada y precisa medición de cada uno de los eslabones de acuerdo con el diseño físico del robot. Es conveniente iniciar con el diseño de la base del robot para luego continuar con el resto de piezas.
- ❖ En cuanto al proceso de exportación es conveniente la instalación previa de la herramienta SimMechanics Link, para limitar la presencia de fallos en el sistema.
- ❖ Los archivos del ensamblaje deben ser importados a MATLAB, guardando los ficheros en una carpeta de trabajo, para posteriormente ejecutar el comando mech-import para la obtención de un sistema en 3D del robot en Simulink análogo al modelo exportado desde Solidworks.
- ❖ Con la prueba de funcionamiento realizada se recomienda la utilización del algoritmo Denavit – Hartenberg para el control de los movimientos de robot y la librería Peter Corke, para el cálculo de la cinemática directa e inversa.
- ❖ Comprobados los movimientos del robot en tiempo real, se considera que los elementos y las características de este estudio son adecuados para ser orientar a futuras investigaciones sobre el control cinemático de los movimientos de un robot y en particular del brazo robótico Kawasaki RS003N.

TRABAJO FUTURO

El modelo de control que se expone en el presente trabajo de investigación constituye la cimiento para futuras investigaciones sobre el control dinámico y de trayectorias.

BIBLIOGRAFÍA

- Almeida, G. (2009). *Fundamentos Generales de Robótica*. Ambato: Universidad Técnica de Ambato.
- Arechavaleta, Y. (2014). *Introducción a la Simulación*. Madrid.
- Barrientos, A., Peñín, L., & Balaguer, R. (1995). *Fundamentos de Robótica*. MacGraw Hill.
- Bonilla, A. (2003). *Guía Tecnopyme. Fase II. Herreamientas de Diseño e Ingeniería*. Zamudio.
- Gómez, O. (2005). *Representación visual de modelos robóticos para aplicaciones didácticas*. Bucaramanga: Universidad Industrial de Santander.
- González, P., & Jimenez, M. (2008). *Robótica Industrial*. Madrid: Instituto de Robótica Industrial.
- González, P., Pineda, M., & Soberanes, D. (2007). *Seguimiento Adaptativo de Trayectorias con convergencia en tiempo finito de un Robot Antropomórfico de Tres grados de Libertad*. Pachuca: Universidad Autónoma del Estado de Hidalgo.
- Infante, J., & Ortiz, U. (2004). *Metodología científica*. México Distrito Federal: Compañía Editorial Continental.
- Kawasaki, R. (23 de 05 de 2016). *Kawasaki Robotics Home*. Recuperado el 23 de 05 de 2016, de <https://robotics.kawasaki.com/en1/products/robots/small-medium-payloads/RS003N/>
- MathWorks. (23 de 05 de 2016). *Accelerating the pace of engineering and science*. Recuperado el 23 de 05 de 2016, de <http://www.mathworks.com/company/newsletters/articles/modeling-complex-mechanical-structures-with-simmechanics.html>
- Matlab. (23 de 05 de 2016). *Math Works*. Recuperado el 23 de 05 de 2016, de <http://es.mathworks.com/products/matlab/>
- Real Academia de la Lengua. REA (2000). *Diccionario de la Lengua Española*. Madrid: Real Académia Española.
- Ecuador. Secretaria Nacional de Planificación y Desarrollo. SENPLADES (2013). *Plan Nacional del Buen Vivir 2013-2017*. Quito.

ANEXOS

ANEXO A: MODELADO CINEMÁTICO DE ROBOT KAWASAKI RS 03N EN MATLAB

```
function varargout = kawa(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @kawa_OpeningFcn, ...
                  'gui_OutputFcn',  @kawa_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
handles.output = hObject;
guidata(hObject, handles);
set(handles.sql, 'Value', 0)
set(handles.sql, 'max', 160)
set(handles.sql, 'min', -160)
set(handles.sql2, 'Value', 0)
set(handles.sql2, 'max', 60) %cambio de signo
set(handles.sql2, 'min', -150) %
set(handles.sql3, 'Value', 0)
set(handles.sql3, 'max', 150)
set(handles.sql3, 'min', -120)
set(handles.sql4, 'Value', 0)
set(handles.sql4, 'max', 360)
set(handles.sql4, 'min', -360)
set(handles.sql5, 'Value', 0)
set(handles.sql5, 'max', 135)
set(handles.sql5, 'min', -135)
set(handles.sql6, 'Value', 0)
set(handles.sql6, 'max', 360)
```



```

set(handles.sq6,'min',-360)
set(handles.spz,'Value',-16)
set(handles.spz,'max',0)
set(handles.spz,'min',-16)
function fcncrearrobot(q1,q2,q3,q4,q5,q6)
cla reset
L(1) = Link([0 0.25 0.12 pi/2 0]);
L(2) = Link('revolute','offset',pi/2, 'd', 0, 'a', 0.25, 'alpha', 0);
L(3) = Link('revolute','offset',-pi/2, 'd', 0, 'a', 0, 'alpha', -pi/2);
L(4) = Link([0 0.25 0 pi/2 0]);
L(5) = Link([0 0 0 -pi/2 0]);
L(6) = Link([0 0.08 0 0 0]);

function fcnmoverrobot()
    global limpiador q1 q2 q3 q4 q5 q6
    pos=[q1, q2, q3,q4,q5,q6];
    limpiador.plot(pos)
    find_system('Robot_KAWASAKI_RS003N')
    set_param('Robot_KAWASAKI_RS003N/Constant','Value',num2str(q1))
    set_param('Robot_KAWASAKI_RS003N/Constant1','Value',num2str(q2))
    set_param('Robot_KAWASAKI_RS003N/Constant2','Value',num2str(q3))
    set_param('Robot_KAWASAKI_RS003N/Constant3','Value',num2str(q4))
    set_param('Robot_KAWASAKI_RS003N/Constant4','Value',num2str(q5))
    set_param('Robot_KAWASAKI_RS003N/Constant5','Value',num2str(q6))
    function sq2_Callback(hObject, eventdata, handles)
    global limpiador q1 q3 q4 q5 q6 q2
    q2=deg2rad(get(handles.sq2,'Value'));
    set(handles.vsq2,'String',rad2deg(q2));
    set(handles.etvsq2,'String',rad2deg(q2));
    fcnmoverrobot()
    pos=[q1 q2 q3 q4 q5 q6]
    CD=fkine(limpiador,pos)
    set(handles.etx,'string',CD(1,4) );
    set(handles.ety,'string',CD(2,4) );
    set(handles.etz,'string',CD(3,4) );
    function pushbutton5_Callback(hObject, eventdata, handles)
    set_param('Robot_KAWASAKI_RS003N','SimulationCommand','start');
    %=====CONTROL DE ANGULOS=====
    while ( igrados(1,1)>=360 || igrados(1,1)<=-360 )

```

```

        if( igrados(1,1)>0)
            igrados(1,1)= igrados(1,1)-360;
        end
    if( igrados(1,1)<0)
        igrados(1,1)= igrados(1,1)+360;
    end
end
    while ( igrados(1,2)>=360 || igrados(1,2)<=-360 )
        if( igrados(1,2)>0)
            igrados(1,2)= igrados(1,2)-360;
        end
    if( igrados(1,2)<0)
        igrados(1,2)= igrados(1,2)+360;
    end
end
    while ( igrados(1,3)>=360 || igrados(1,3)<=-360 )
        if( igrados(1,3)>0)
            igrados(1,3)= igrados(1,3)-360;
        end
    if( igrados(1,3)<0)
        igrados(1,3)= igrados(1,3)+360;
    end
end
    while ( igrados(1,4)>=360 || igrados(1,4)<=-360 )
        if( igrados(1,4)>0)
            igrados(1,4)= igrados(1,4)-360;
        end
    if( igrados(1,4)<0)
        igrados(1,4)= igrados(1,4)+360;
    end
end
    while ( igrados(1,5)>=360 || igrados(1,5)<=-360 )
        if( igrados(1,5)>0)
            igrados(1,5)= igrados(1,5)-360;
        end
    if( igrados(1,5)<0)
        igrados(1,5)= igrados(1,5)+360;
    end
end

```

```

end
while ( igrados(1,6)>=360 || igrados(1,6)<=-360 )
    if( igrados(1,6)>0)
        igrados(1,6)= igrados(1,6)-360;
    end
end
if( igrados(1,6)<0)
    igrados(1,6)= igrados(1,6)+360;
end
end

set(handles.vsq1,'string',round(igrados(1,1)) );
set(handles.vsq2,'string',round(igrados(1,2)) );
set(handles.vsq3,'string',round(igrados(1,3)) );
set(handles.vsq4,'string',round(igrados(1,4)) );
set(handles.vsq5,'string',round(igrados(1,5)) );
set(handles.vsq6,'string',round(igrados(1,6)) );
set(handles.etvsq1,'string',round(igrados(1,1)) );
set(handles.etvsq2,'string',round(igrados(1,2)) );
set(handles.etvsq3,'string',round(igrados(1,3)) );
set(handles.etvsq4,'string',round(igrados(1,4)) );
set(handles.etvsq5,'string',round(igrados(1,5)) );
set(handles.etvsq6,'string',round(igrados(1,6)) );
posq1=deg2rad(str2double(get(handles.etvsq1,'string')));
posq2=deg2rad(str2double(get(handles.etvsq2,'string')));
posq3=deg2rad(str2double(get(handles.etvsq3,'string')));
posq4=deg2rad(str2double(get(handles.etvsq4,'string')));
posq5=deg2rad(str2double(get(handles.etvsq5,'string')));
posq6=deg2rad(str2double(get(handles.etvsq6,'string')));
q1=posq1;
q2=posq2;
q3=posq3;
q4=posq4;
q5=posq5;
q6=posq6;
if (op3==1)
    q=(0:1:25);
    traj=jtraj(CI2,CI,q)
end

```

ANEXO B: DIAGRAMA DE FLUJO DE LA SIMULACIÓN

